AFRL-RI-RS-TR-2018-098

LARGE-SCALE PARAPHRASING FOR NATURAL LANGUAGE UNDERSTANDING

JOHNS HOPKINS UNIVERSITY

APRIL 2018

FINAL TECHNICAL REPORT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

STINFO COPY

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE

■ **AIR FORCE MATERIEL COMMAND** ■ **UNITED STATES AIR FORCE** ■ **ROME, NY 13441**

NOTICE AND SIGNATURE PAGE

AFRL-RI-RS-TR-2018-098 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE CHIEF ENGINEER:

/ S /	**/ S /**
WALTER V. GADZ JR	JON S. JONES
Work Unit Manager	Technical Advisor, Information Intelligence Systems and Analysis Division Information Directorate

REPORT DOCUMENTATION PAGE	*Form Approved* **OMB No. 0704-0188**

1. REPORT DATE *(DD-MM-YYYY)* APRIL 2018	**2. REPORT TYPE** FINAL TECHNICAL REPORT	**3. DATES COVERED** *(From - To)* OCT 2012 – NOV 2017

4. TITLE AND SUBTITLE LARGE-SCALE PARAPHRASING FOR NATURAL LANGUAGE UNDERSTANDING	**5a. CONTRACT NUMBER** FA8750-13-2-0017
	5b. GRANT NUMBER N/A
	5c. PROGRAM ELEMENT NUMBER 62303E
6. AUTHOR(S) Chris Callison-Burch, Benjamin Van Durme	**5d. PROJECT NUMBER** DEFT
	5e. TASK NUMBER 12
	5f. WORK UNIT NUMBER 07
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Johns Hopkins University 3400 N. Charles Street Baltimore MD 21218	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Air Force Research Laboratory/RIEA 525 Brooks Road Rome NY 13441-4505	**10. SPONSOR/MONITOR'S ACRONYM(S)** AFRL/RI
	11. SPONSOR/MONITOR'S REPORT NUMBER AFRL-RI-RS-TR-2018-098

12. DISTRIBUTION AVAILABILITY STATEMENT
Approved for Public Release; Distribution Unlimited. This report is the result of contracted fundamental research deemed exempt from public affairs security and policy review in accordance with SAF/AQR memorandum dated 10 Dec 08 and AFRL/CA policy clarification memorandum dated 16 Jan 09

13. SUPPLEMENTARY NOTES

14. ABSTRACT

In this project, we researched and developed technologies to automatically extract large-volumes of paraphrases to aid in natural language understanding (NLU) tasks. We developed three core algorithms to: (1) generate extremely large paraphrase databases, and (2) adapt paraphrase databases to new domains, and (3) augment paraphrase rules with fine-grained semantic entailment relations. Our work introduced the paraphrase database (PPDB), the largest paraphrase resource developed to date. The resource contains over 100 million paraphrases for English. We generated paraphrase databases for 23 foreign languages.

15. SUBJECT TERMS

Paraphrase, knowledge base, FrameNet, semantic entailment, graph clustering.

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** **WALTER V. GADZ JR.**
a. REPORT U	**b. ABSTRACT** U	**c. THIS PAGE** U	SAR	71	**19b. TELEPHONE NUMBER (*Include area code*)** **N/A**

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. SUMMARY

In this project, our team headed by Benjamin Van Durme at Johns Hopkins University and Chris Callison-Burch at the University of Pennsylvania developed methods to automatically extract large-volumes of paraphrases to aid in natural language understanding (NLU) tasks. We developed three core algorithms to: (1) generate extremely large paraphrase databases, and (2) adapt paraphrase databases to new domains, and (3) augment paraphrase rules with fine-grained semantic entailment relations. Our work introduced the paraphrase database (PPDB), the largest paraphrase resource developed to date. The resource contains over 100 million paraphrases for English, including single word synonyms or lexical paraphrases like (*jailed, incarcerated)*, paraphrases where one word rewrites as many words like (*jailed, held in prison)*, phrasal paraphrases like *(placed in detention, taken into custody)*, and syntactic rewrite rules like ([NP$_1$] *arrested* [NP$_2$], [NP$_2$] *was arrested by* [NP$_1$]). We performed a substantial engineering effort to extract paraphrases from large volumes of data, and released our tools for doing so as part of an open source package. We used these tools to generate an English paraphrase database, as well as paraphrase databases for 23 different languages: Arabic, Bulgarian, Chinese, Czech, Dutch, Estonian, Finnish, French, German, Greek, Hungarian, Italian, Latvian, Lithuanian, Polish, Portuguese, Romanian, Russian, Slovak, Slovenian, and Swedish. We introduced a variety of techniques for sorting the automatically extracted paraphrases so that they are ranked similarly to human judgments of paraphrase quality. We automatically labeled every paraphrase pair in PPDB with a semantic entailment relation. This lightweight semantics allows our paraphrases to be used for some textual inference tasks that are an important part of knowledge base population (KBP). We further refined the semantics of our paraphrase resource by clustering paraphrases by word sense.

Highlights of this work include:

- This project had 27 publications in top-tier conferences, plus 7 peer reviewed workshop publications and 2 PhD theses. 7 of these publications have over 50 citations as of February 2018, including the PPDB paper, which has nearly 300 citations.
- The release of the paraphrase database fueled a great amount of research performed by ourselves and other groups on improving the quality of word embeddings by altering their vector representations to more closely mirror the paraphrases in PPDB.
- This work also significantly advanced development of the Joshua machine translation toolkit, which is now used by the DoD.
- We advanced the state of the art in NLU through data-driven paraphrasing, and performed a variety of intrinsic evaluations to quantify the contributions of methodological advances. We show how PPDB can be used to expand the coverage of hand crafted lexical-semantic resources like FrameNet.

2 INTRODUCTION

Paraphrases are alternative ways of expressing the same information. Automatically generating and detecting paraphrases is a crucial aspect of many NLP tasks. In multi-document summarization, paraphrase detection is used to collapse redundancies. Paraphrase generation can be used for query expansion in information retrieval and question answering systems. Paraphrases allow for more flexible matching of system output against human references for tasks like machine translation and automatic summarization. In KBP they can be used to help map from the many ways that it is possible to express a proposition in natural language onto the relation in the KB. For example, we would like to construct a single subgraph in the KB like the one below:

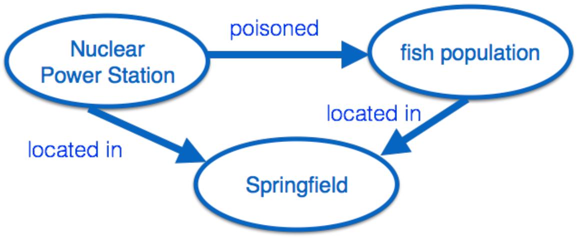


Figure 1: In a knowledge base, we want to have a single representation that expresses the relationship between entities.

In natural language, there are many possible ways of expressing the same information. For instance, all of the following sentences could convey the information.

Springfield's nuclear power station contaminated local fish populations
Atomic power generation in Springfield polluted indigenous seafood stocks
Radioactive power generation tainted Springfield's municipal fishing resources
Regional salmon stocks were poisoned by Springfield's nuclear plant

Figure 2: In natural language, there are many possible ways of expressing the same information.

One of the goals of learning paraphrases is to be able to recognize the many possible ways of expressing the same information. The table below shows PPDB's paraphrases for the words in the top row.

Table 1: Example paraphrases drawn from PPDB.

Springfield's	**nuclear**	**power**	**plant**	**contaminated**	**local**	**fish**	**populations**
	nuclear power station nuclear plant power plant					fish stocks stocks fishery resources fishing resources resources	
		power station generating station power generation					
	atomic radioactive fissile nuclear-related	energy authority electricity wattage electric	factory station facility	infected polluted tainted poisoned impacted affected sullied afflicted exposed tarnished	domestic local-level municipal indigenous localized regional	fishes fishing fisheries catch seafood	stocks residents inhabitants communities groups dwellers

As part of our DEFT project we released the paraphrase database, called PPDB for short. PPDB was trained from bilingual parallel corpora between English and 22 other languages, totaling 106 million sentence pairs and a total of 2 billion English tokens.

PPDB contains 8 million synonyms, 3 million one-to-many paraphrases, 68 million phrase-to-phrase paraphrases, and 94 million meaning-preserving syntactic transformations (representing linguistic phenomena like the English possessive rule, dative shift, the partitive construction, and many others). PPDB is freely available from our web site paraphrase.org. It is a much larger resource than the manually-constructed WordNet resource that is heavily used in NLP research, and much larger than other past automatically-generated paraphrase resources like DIRT and the Microsoft Research Paraphrase Corpus. We also released a multilingual PPDB that includes a collection of paraphrases in 23 different languages: Arabic, Bulgarian, Chinese, Czech, Dutch, Estonian, Finnish, French, German, Greek, Hungarian, Italian, Latvian, Lithuanian, Polish, Portuguese, Romanian, Russian, Slovak, Slovenian, and Swedish. Example Spanish paraphrases for *estupefacientes* are *narcóticos, drogas, droga, narcotráfico, medicamentos,* and *fármacos.* Example Chinese paraphrases for上任 are 上台, 就职, 就任, 出任, 执政, 办事处, 前, and 办公室.

Our method for automatically extracting paraphrases from data, builds on the idea of bilingual pivoting. We extract paraphrases from bilingual parallel corpora by identifying equivalent English expressions using a shared foreign phrase. This ensures that their meaning is similar. Figure 3 illustrates the method. *Thrown into jail* occurs many times in the training data, aligning with several different foreign phrases. Each of these may align with a variety of other English paraphrases. Thus, *thrown into jail* not only paraphrases as *imprisoned*, but also as *arrested, detained, incarcerated, jailed, locked up, taken into custody*, and *thrown into prison*. However, not all the paraphrases are uniformly good. The baseline method also extracts candidate paraphrases that mean the same thing, but do not share the same syntactic category as the original phrase, such as *be thrown in prison, been thrown into jail, being arrested, in jail, in prison, put in prison for, were thrown into jail*, and *who are held in detention*. It is also prone to generating many bad paraphrases, such as *maltreated, thrown, cases, custody, arrest, owners*, and *protection*, because of noisy/inaccurate word alignments and other problems. Separating good paraphrases from bad presents important research challenges, which we also addressed during the DEFT program.

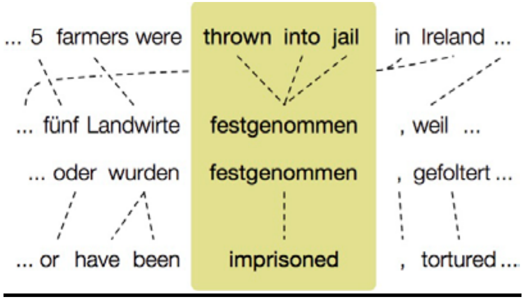
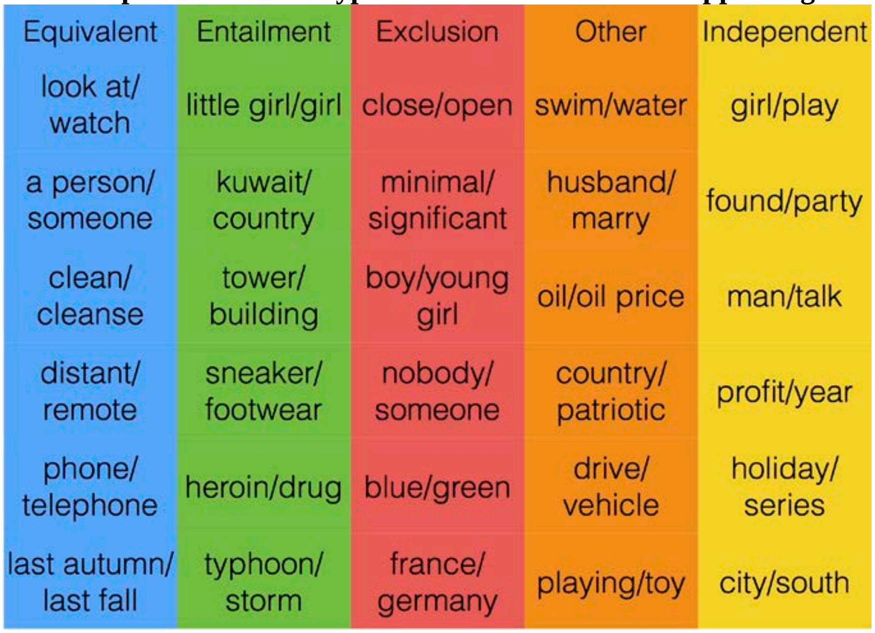


Figure 3: The German *festgenommen* links the English phrase thrown into jail to its paraphrase imprisoned

In addition to extracting paraphrases, we added an interpretable semantics to PPDB. Rather than defining the relationship between the phrase pairs in the database simply as "approximately equivalent". Our research allowed these pairs to be assigned more nuanced semantic relations, including directed entailment (*little girl/girl*) and exclusion (*nobody/someone*). We automatically assigned semantic entailment relations to all 100+ million entries in PPDB using features derived from past work on discovering inference rules from text and semantic taxonomy induction. Examples are given in Table 2.

Table 2: Examples of different types of entailment relations appearing in PPDB.



Equivalent	Entailment	Exclusion	Other	Independent
look at/ watch	little girl/girl	close/open	swim/water	girl/play
a person/ someone	kuwait/ country	minimal/ significant	husband/ marry	found/party
clean/ cleanse	tower/ building	boy/young girl	oil/oil price	man/talk
distant/ remote	sneaker/ footwear	nobody/ someone	country/ patriotic	profit/year
phone/ telephone	heroin/drug	blue/green	drive/ vehicle	holiday/ series
last autumn/ last fall	typhoon/ storm	france/ germany	playing/toy	city/south

In addition to assigning entailment relations, we also partitioned paraphrases in PPDB into groups of WordNet-like synsets. Instead of noun *bug* would yield a single list of paraphrases that includes *insect, glitch, beetle, error, microbe, wire, cockroach, malfunction, microphone, mosquito, virus, tracker, pest, informer, snitch, parasite, bacterium, fault, mistake, failure*, we cluster these into word senses as shown in Figure 4.

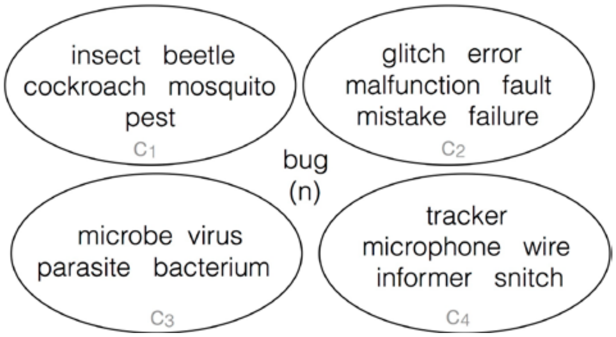


Figure 4: The word bug has several distinct meanings. Here we automatically cluster its paraphrases into groups that correspond to those different meanings.

3 METHODS, ASSUMPTIONS AND PROCEDURES

3.1 Paraphrase Extraction

To extract paraphrases, we follow Bannard and Callison-Burch (2005)'s bilingual pivoting method. The intuition is that two English strings *e1* and *e2* that translate to the same foreign string *f* can be assumed to have the same meaning. We can thus *pivot* over *f* and extract <*e1*,*e2*> as a pair of paraphrases, as illustrated in Figure 3. The method extracts a diverse set of paraphrases. For *thrown into jail*, it extracts *arrested*, *detained*, *imprisoned*, *incarcerated*, *jailed*, *locked up*, *taken into custody*, and *thrown into prison*, along with a set of incorrect/noisy paraphrases that have different syntactic types or that are due to misalignments.

For PPDB, we formulate our paraphrase collection as a weighted *synchronous context-free grammar* (SCFG) (Aho and Ullman, 1972; Chiang, 2005) with syntactic nonterminal labels, similar to Cohn and Lapata (2008) and Ganitkevitch et al. (2011). An SCFG rule has the form:

$$r \overset{\text{def}}{=} C \rightarrow < f, e, \sim, \overrightarrow{\varphi} > \qquad \textbf{(1)}$$

where the left-hand side of the rule, *C*, is a nonterminal and the right-hand sides *f* and *e* are strings of terminal and nonterminal symbols. There is a one-to-one correspondence, ~, between the nonterminals in *f* and *e*: each nonterminal symbol in *f* has to also appear in *e*. Each rule r is annotated with a vector of feature functions $\vec{\varphi} = \{\varphi_1 \dots \varphi_N\}$ which are combined in a log-linear model (with weights $\vec{\lambda}$) to compute the *cost* of applying *r*:

$$cost(r) = -\sum_{i=1}^{n} \lambda_i log\varphi_i \qquad \textbf{(2)}$$

To create a syntactic paraphrase grammar, we first extract a foreign-to-English translation grammar from a bilingual parallel corpus, using techniques from syntactic machine translation (Koehn, 2010). Then, for each pair of translation rules where the left-hand side *C* and foreign string *f* match:

$$r_1 \overset{\text{def}}{=} C \rightarrow < f, e_1, \sim_1, \overrightarrow{\varphi_1} > \qquad \textbf{(3)}$$

$$r_2 \overset{\text{def}}{=} C \rightarrow < f, e_2, \sim_2, \overrightarrow{\varphi_2} > \qquad \textbf{(4)}$$

we *pivot* over *f* to create a paraphrase rule r_p:

$$r_p \overset{\text{def}}{=} C \rightarrow < e_1, e_2, \sim_p, \overrightarrow{\varphi_p} > \qquad \textbf{(5)}$$

with a combine nonterminal correspondency function $\sim_p$. Note that the common source fide *f* implies that $e_1$and $e_2$share the same set of nonterminal symbols.

The paraphrase rules obtained using this method are capable of making well-formed generalizations of meaning-preserving rewrites in English. For instance, we can combine two French-English translation rules

$$NP \rightarrow < NP\,'s\; NN, \qquad le\; NN\; de\; NP >$$

$$NP \rightarrow < the\; NN\; of\; NP, le\; NN\; de\; NP >$$

to extract the following English paraphrase:

$$NP \rightarrow < the\; NN\; of\; NP, NP\,'s\; NN >$$

This captures the English possessive rule which is a general re-write rule that allows us to transform expressions like *the screen of the laptop* into *the laptop's screen*, where the two noun elements of the phrase are reordered with respect to each other. A wide variety of meaning preserving syntactic transformations are captured in the paraphrase rules that we extract via pivoting over syntactic translation rules. Table 3 shows a variety of the re-write rules that are well-known to linguists, with examples of how they are realized in our paraphrase database.

Table 3: Examples of some of the meaning preserving syntactic transformations that are found in PPDB.

Possessive rule	NP →	the NN of the NNP	the NNP’s NN
	NP →	the NNS$_1$ made by the NNS$_2$	the NNS$_2$’s NNS$_1$
Dative shift	VP →	give NN to NP	give NP the NN
	VP →	provide NP$_1$ to NP$_2$	give NP$_2$ NP$_1$
Adv. | adj. phrase move	S →	ADVP they VBD	they VBD ADVP
	S →	it is ADJP VP	VP is ADJP
Verb particle shift	VP →	VB NP up	VB up NP
Reduced relative clause	S →	although PRP VBD that	although PRP VBD
	ADJP →	very JJ that S	JJ S
Partitive constructions	NP →	CD of the NN	CD NN
	NP →	all DT\NP	all of the DT\NP
Topicalization	S →	NP, VP.	VP, NP.
Passivization	SBAR →	that NP had VBN	which was VBN by NP
Light verbs	VP →	take action ADVP	to act ADVP
	VP →	to make a decision PP	to decide PP

3.2 Data Used to Construct the Paraphrase Database

We aggregated several English-to-foreign bilingual parallel corpora to extract PPDB: Europarl v7 (Koehn, 2005), consisting of bitexts for the 19 European languages, the 10^9 French-English corpus (Callison-Burch et al., 2009), the Czech, German, Spanish and French portions of the News Commentary data (Koehn and Schroeder, 2007), the United Nations French- and Spanish-English parallel corpora (Eisele and Chen, 2010), the JRC Acquis corpus (Steinberger et al., 2006), Chinese and Arabic newswire corpora used for the GALE machine translation campaign, parallel Urdu-English data from the NIST translation task, the French portion of the OpenSubtitles corpus (Tiedemann, 2009), and a collection of Spanish-English translation memories provided by TAUS. The resulting composite parallel corpus had more than 106 million sentence pairs, over 2 billion English words, and spans 22 pivot languages.

3.3 Paraphrase Scores

Each of the paraphrase entries in PPDB has a set of associated feature functions, which are stored in the paraphrase feature vector $\overrightarrow{\varphi_p}$. These may be useful for ranking the quality of the paraphrases themselves. For instance, Zhao et al. (2008) proposed a log-linear model for scoring paraphrases instead of Bannard and Callison-Burch's paraphrase probability. Malakasiotis and Androutsopoulos (2011) re-ranked paraphrases using a maximum entropy classifier and a support vector regression ranker to set weights for features associated with a set of paraphrases, optimizing to a development set that was manually labeled with quality scores.

Table 4: An example paraphrase rule from the PPDB. The six fields are the left-hand side nonterminal, the phrase, the paraphrase, the features associated with the rule, the word-alignment correspondence between the phrase and the paraphrase, and the predicted

```
[NN] ||| incarceration ||| imprisonment ||| PPDB2.0Score=4.02725 PPDB1.0Score=5.831490 -logp(LHS|e1)=0.03320 -
logp(LHS|e2)=0.04620 -logp(e1|LHS)=12.31856 -logp(e1|e2)=4.26445 -logp(e1|e2,LHS)=3.97651 -logp(e2|LHS)=9.63415 -
logp(e2|e1)=1.56704 -logp(e2|e1,LHS)=1.29210 AGigaSim=0.65317 Abstract=0 Adjacent=0 CharCountDiff=-1
CharLogCR=-0.08004 ContainsX=0 Equivalence=0.427150 Exclusion=0.000101 GlueRule=0 GoogleNgramSim=0.04294
Identity=0 Independent=0.078898 Lex(e1|e2)=59.79539 Lex(e2|e1)=59.79539 Lexical=1 LogCount=4.20469
MVLSASim=NA Monotonic=1 OtherRelated=0.368458 PhrasePenalty=1 RarityPenalty=0 ReverseEntailment=0.125394
SourceTerminalsButNoTarget=0 SourceWords=1 TargetComplexity=0.99921 TargetFormality=1.00000
TargetTerminalsButNoSource=0 TargetWords=1 UnalignedSource=0 UnalignedTarget=0 WordCountDiff=0 WordLenDiff=-
1.00000 WordLogCR=0  ||| 0-0 ||| Equivalence
```

Table 4 gives an example paraphrase rule for English. The entry contains 4 fields separated by |||. The first field is the left-hand side (LHS) nonterminal symbol that dominates the SCFG rule. The second field is the original phrase (which can be a mix of words and nonterminal symbols). The third field is the paraphrase. If the paraphrase is a syntactic rule it will have an identical set of nonterminal symbols as the original phrase, but they can appear in different orders. The mapping between nonterminal symbols is given with indices like [NP ,1] and [NP,2]. The fourth field is a collection of features associated with the rule.

The features we estimate for each paraphrase rule are related to features typically used in machine translation systems. Features that contain probability estimates, like $p(e_2|e_1)$, are stored as their negative logarithm $-\log p(e_2|e_1)$. The features we compute for each PPDB rule are logically grouped into the following sets:

3.3.1 Paraphrase probability scores

Our paraphrase probability scores are inspired by Bannard and Callison-Burch's original work on extracting paraphrases from bilingual parallel corpora. They defined a paraphrase probability as

$$p(e_2|e_1) \approx \sum_f p(e_2|f)\, p(f|e_1) \qquad \textbf{(6)}$$

We encode this score as a feature on each paraphrase rule:

- **p(e2|e1)** – the paraphrase probability of the paraphrase given the original phrase, This is given as a negative log value.
- **p(e1|e2)** – the paraphrase probability of the original phrase given the paraphrase. This is given as a negative log value.

We additionally include two related paraphrase probabilities:

- **Lex(e2|e1)** – the lexicalized paraphrase probability of the paraphrase given the original phrase. This feature is estimated as defined by Koehn et al. (2003).
- **Lex(e1|e2)** – the lexicalized paraphrase probability of phrase given the paraphrase.

These are the average of the $p(e_2|e_1)$ scores for individual words within the phrasal paraphrases. The word-alignment correspondences are given by the fifth field in the paraphrase rule, like 0-0 in Table 4.

3.3.2 Monolingual Distributional Similarity Scores

The bilingual pivoting approach anchors paraphrases that share an interpretation because of a shared foreign phrase. Paraphrasing methods based on monolingual text corpora, like DIRT (Lin and Pantel, 2001), measure the similarity of phrases based on distributional similarity. This results in a range of different types of phrases, including paraphrases, inference rules and antonyms. For instance, for *thrown into prison* DIRT extracts good paraphrases like arrested, *detained,* and *jailed*. However, it also extracts phrases that are temporarily or causally related like *began the trial of, cracked down on, interrogated, prosecuted* and *ordered the execution of*, because they have similar distributional properties. Since bilingual pivoting rarely extracts these non-paraphrases, we can use monolingual distributional similarity to re-rank paraphrases extracted from bitexts (following Chan et al. (2011)) or incorporate a set of distributional similarity scores as features in our log-linear model.

Each similarity score relies on precomputed distributional signatures that describe the contexts that a phrase occurs in. To describe a phrase *e*, we gather counts for a set of contextual features for each occurrence of *e* in a corpus. Writing the context vector for the *i*-th occurrence of *e* as $\overrightarrow{s_{e,\iota}}$, we can aggregate over all occurrence of *e*, resulting in a distributional signature for e, $\overrightarrow{s_e} = \sum_i \overrightarrow{s_{e,\iota}}$. Following the intuition that phrases with similar meanings occur in similar contexts, we can then quantify the goodness of *e2* as a paraphrase of *e1* by computing the cosine similarity between their distributional signatures:

$$sim(e_1, e_2) = \frac{\overrightarrow{s_{e1}} \cdot \overrightarrow{s_{e2}}}{|\overrightarrow{s_{e1}}||\overrightarrow{s_{e2}}|} \qquad \textbf{(7)}$$

A wide variety of features have been used to describe the distributional context of a phrase. Rich, linguistically informed feature-sets that rely on dependency and constituency parses, part-of-speech tags, or lemmatization have been proposed in work such as by Church and Hanks (1991) and Lin and Pantel (2001). For instance, a phrase is described by the various syntactic relations such as: “what verbs have this phrase as the subject?”, or “what adjectives modify this phrase?”. Other work has used simpler n-gram features, e.g. “what words or bigrams have we seen to the left of this phrase?”. A substantial body of work has focused on using this type of feature-set for a variety of purposes in NLP (Lapata and Keller, 2005; Bhagat and Ravichandran, 2008; Lin et al., 2010; Van Durme and Lall, 2010).

For PPDB, we compute n-gram-based context signatures for the 200 million most frequent phrases in the Google n-gram corpus (Brants and Franz, 2006; Lin et al., 2010), and richer linguistic signatures for 175 million phrases in the Annotated Gigaword corpus (Napoles et al., 2012). Our features are:

- **GoogleNgramSim -** n-gram based features for words seen to the left and right of a phrase.
- **AGigaSim** - Position-aware lexical, lemma-based, part-of-speech, and named entity class unigram and bigram features, drawn from a three-word window to the right and left of the phrase. Incoming and outgoing (wrt. the phrase) dependency link features, labeled with the corresponding lexical item, lemmata and POS.

- **MVLSASim -** Multiview LSA similarity. This is a generalization of Latent Semantic Analysis (LSA) that supports the fusion of arbitrary views of data and relies on Generalized Canonical Correlation Analysis (Rastorgi et al, 2015)

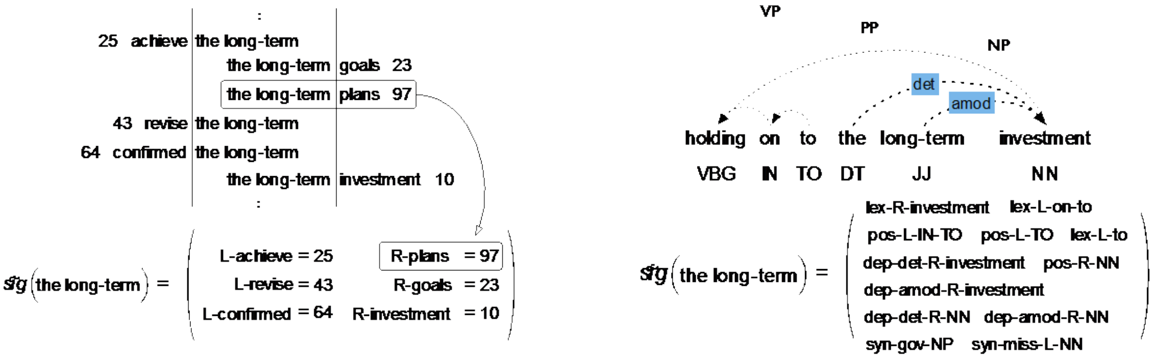
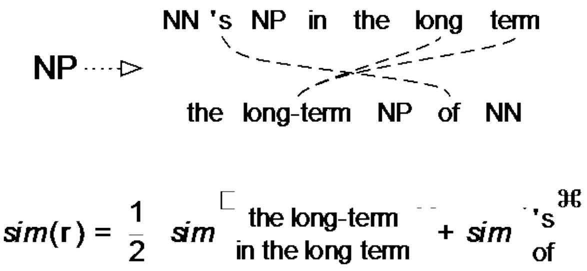


Figure 5: Features extracted for the phrase *the long term* from the n-gram corpus (left) and Annotated Gigaword (right)

Figure 5 gives an illustration of how we compute the vector signatures for phrases using n-gram derived features and syntactic annotations. The n-gram corpus records *the long-term* as preceded by *revise* (43 times), and followed by *plans* (97 times). We add corresponding features to the phrase's distributional signature retaining the counts of the original n-grams. For the AGigaSim feature, we a context vector containing position-aware lexical and part-of-speech n-gram features, labeled dependency links, and features reflecting the phrase's CCG-style syntactic label NP/NN.

The syntactic paraphrase rules contain non-terminal symbols, which makes computing their distributional similarity non-trivial. So that we were able to compute **GoogleNgramSim, AGigaSim** and **MVLSASim** for these kinds of paraphrases, we performed a word alignment on the terminal symbols in the rules, and then computed the average of similarity for each aligned n-gram sequence within the rule, as shown in Figure 6.



NP
NN 's NP in the long term
the long-term NP of NN
sim(r) = 1/2 sim the long-term / in the long term + sim 's / of

Figure 6: To compute the monolingual distributional similarity for syntactic paraphrase rules, we performed a word alignment over the terminal symbols, and then averaged the similarity of the aligned phrases.

3.3.3 Syntactic features

We derived syntactic features for any constituents governing the phrase. These features include:

- **p(LHS|e2)** – the (negative log) probability of the left-hand side nonterminal symbol given the paraphrase.
- **p(LHS|e1)** – the (negative log) probability of the left-hand side nonterminal symbol given the original phrase.
- **p(e2|LHS)** – the (negative log) probability of the paraphrase given the left-hand side nonterminal symbol (this is typically a very low probability).
- **p(e2|f1,LHS)** – the (negative log) probability of paraphrase given the lefthand side nonterminal symbol and the original phrase.
- **p(e1|LHS)** – the (negative log) probability of original phrase given the lefthand side nonterminal (this is typically a very low probability).
- **p(e1|e2,LHS)** – the (negative log) probability of original phrase given the lefthand side nonterminal symbol and the paraphrase.

3.3.4 Semantic entailment features

We assign each paraphrase entry to a semantic entailment class. The entailment class with the maximum probability is given as the sixth feature in the paraphrase rule. For example, in Table 4 the entailment class for the paraphrase <*incarceration, imprisonment*> is *Equivalence.* We also provide features that give a probability distribution over all of our semantic entailment classes:

- **Equivalence –** the probability that the paraphrase pair stands in an equivalence relationship
- **Exclusion –** the probability that the phrase and paraphrase are mutually exclusive of one another
- **Forward Entailment –** the probability that the phrase entails the paraphrase
- **Independent –** the probability that the phrase and paraphrase are unrelated to one another
- **Reverse Entailment –** the probability that the paraphrase entails the phrase
- **OtherRelated –** the probability that the phrase and paraphrase stand in some other sort of semantic relationship (like <*Israel, Israeli*> or <*plane, sky*>)

3.3.5 Features derived from machine translation rules

- **Abstract** – a binary feature that indicates whether the rule is composed exclusively of nonterminal symbols.
- **Adjacent** – a binary feature that indicates whether rule contains adjacent nonterminal symbols.
- **ContainsX** – a binary feature that indicates whether the nonterminal symbol X is used in this rule. X is the symbol used in Hiero grammars (Chiang, 2007), and is sometimes used by our syntactic SCFGs when we are unable to assign a linguistically motivated nonterminal.
- **GlueRule** – a binary feature that indicates whether this is a glue rule. Glue rules are treated specially by the Joshua decoder (Post et al., 2013). They are used when the decoder cannot produce a complete parse using the other grammar rules.
- **Identity** – a binary feature that indicates whether the phrase is identical to the paraphrase.
- **Lexical** – a binary feature that says whether this is a single word paraphrase.
- **LogCount** – the log of the frequency estimate for this paraphrase pair.
- **Monotonic** – a binary feature that indicates whether multiple nonterminal symbols occur in the same order (are monotonic) or if they are re-ordered.
- **PhrasePenalty** – this feature is used by the decoder to count how many rules it uses in a derivation. Turning helps it to learn to prefer fewer longer phrases, or more shorter phrases. The value of this feature is always 1.

- **RarityPenalty** – this feature marks rules that have only been seen a handful of times. It is calculated as exp(1 − c(e, f)), where c(e, f) is the estimate of the frequency of this paraphrase pair.
- **SourceTerminalsButNoTarget** – a binary feature that fires when the phrase contains terminal symbols, but the paraphrase contains no terminal symbols.
- **SourceWords** – the number of words in the original phrase.
- **TargetTerminalsButNoSource** – a binary feature that fires when the paraphrase contains terminal symbols but the original phrase only contains nonterminal symbols.
- **TargetWords** – the number of words in the paraphrase.
- **UnalignedSource** – a binary feature that fires if there are any words in the original phrase that are not aligned to any words in the paraphrase.
- **UnalignedTarget** – a binary feature that fires if there are any words in the paraphrase that are not aligned to any words in the original phrase.

3.3.6 Miscellaneous features related to our experiments in text-to-text generation

Our lab has been experimenting with using paraphrases to perform text-to-text generation. For example, we looked into re-writing English strings to be shorter (called sentence compression in the NLP literature), and to be simpler (called text simplification). Figure 7 shows an example of how we can re-write a sentence to be shorter using paraphrases.

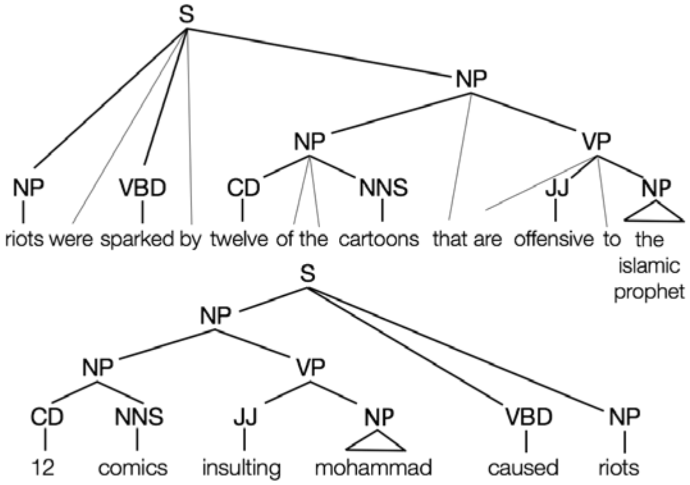

S
NP
NP
VP
NP
VBD
CD
NNS
JJ
NP
riots were sparked by twelve of the cartoons that are offensive to the islamic prophet
S
NP
NP
VP
CD
NNS
JJ
NP
VBD
NP
12
comics
insulting
mohammad
caused
riots

Figure 7: An example of using paraphrases for monolingual text-to-text generation, where a longer sentence can be rewritten to be shorter.

To support these experiments in text-to-text generation, we add a set of paraphrase features that are relevant to those tasks.

- **CharCountDiff** – a feature that calculates the difference in the number of characters between the phrase and the paraphrase. This feature is used for our sentence compression experiments (Napoles et al., 2011).
- **CharLogCR** – the log-compression ratio in characters, $\log(\frac{chars(e2)}{chars(e1)})$, another feature used in sentence compression.

- **WordCountDiff** – the difference in the number of words in the original phrase and the paraphrase. This feature is used for our sentence compression experiments.
- **WordLenDiff** – the difference in average word length between the original phrase and the paraphrase. This feature is useful for text compression and simplification experiments.
- **WordLogCR** – the log compression ratio in words, estimated as $\log(\frac{words(e2)}{words(e1)})$ This feature is used for our sentence compression experiments.
- **TargetComplexity** – A score for how complex the words in paraphrase are (Pavlick and Nenkova 2015)
- **TargetFormality -** A score for how formal the language used in the paraphrase is (Pavlick and Nenkova 2015)

As an example of the **TargetComplexity** score, here are the 15 paraphrases of *the end* sorted from the most complex (according to our automatic score) to the least complex: *the finalization* (rated the most complex), *the expiration, the demise, the completion, the closing, the latter part, termination, goal, the close, late, the final analysis, the last, the finish, the final part, the last part* (rated the simplest).

3.3.7 Features to sort paraphrases

We combine a subset of the features to produce a PPDB 1.0 SCORE feature, that we use to sort the paraphrases.

$$\begin{aligned}\textbf{PPDB 1.0 SCORE} = \; & \mathbf{p(e2|e1) + p(e1|e2) + p(e2|e1,lhs)} \\ & \mathbf{+ \; p(e1|e2,lhs) + 100 \cdot RarityPenalty} \\ & \mathbf{+ \; 0.3 \cdot p(lhs|e2) + 0.3 \cdot p(lhs|e1)} \qquad \mathbf{(8)}\end{aligned}$$

The selection of features and the values for their weights are chosen in an ad hoc fashion, based on our intuitions about which features seem to be useful for sorting higher quality paraphrases from lower quality paraphrases. A more principled is to collect a set of judgments about the quality of a random sample of the paraphrases, and then use logistic regression to fit the weights to the human judgments, which we did in the second release of the database to get the **PPDB 2.0 SCORE** feature.

We provide the full feature set so that users can re-sort the resource to fit native speaker judgments or to fit the needs of a specific NLP task.

3.4 Sorting Paraphrases by Human Judgments

The notion of ranking paraphrases goes back to the original method that PPDB is based on. Bannard and Callison-Burch (2005) introduced the bilingual pivoting method, which extracts *incarcerated* as a potential paraphrase of *put in prison* since they are both aligned to *festgenommen* in different sentence pairs in an English-German bitext. Since *incarcerated* aligns to many foreign words (in many languages) the list of potential paraphrases is long. Paraphrases vary in quality since the alignments are automatically produced and noisy. In order to rank the paraphrases, Bannard and Callison-Burch (2005) defined a paraphrase probability in terms of the translation model probabilities:

$$p(e_2|e_1) \approx \sum_f p(e_2|f)\, p(f|e_1) \qquad \textbf{(9)}$$

3.4.1 Heuristic scoring in PPDB 1.0

Instead of ranking the paraphrases with a single score, Ganitkevitch et al. (2013) expanded the set of scores in PPDB. The rules in PPDB 1.0 were scored using an ad-hoc weighting of seven of these features, given by the **PPDB 1.0 SCORE** (defined above). This heuristic linear combination of scores was used to divide PPDB into six increasingly large sizes– S, M, L, XL, XXL, and XXXL. PPDB-XXXL contains all of the paraphrase rules and has the highest recall, but the lowest average precision. The smaller sizes contain better average scores but offer lower coverage. Ganitkevitch et al. (2013) performed a small-scale analysis of how their heuristic score correlated with human judgments by collecting <2,000 judgments for PPDB paraphrases of verbs that occurred in Propbank.

3.4.2 Supervised scoring model in PPDB 2.0

For the second release of PPDB, we ranked the paraphrases using a supervised scoring model. To train the model, we collected human judgements for 26,455 paraphrase pairs sampled from PPDB. Each paraphrase pair was judged by 5 people who each assigned a score on a 5-point Likert scale, as described in Callison-Burch (2008). These 5 scores were averaged.

We used these human judgments to fit a regression to the 33 features available in the PPDB 1.0 feature vector, plus an additional 176 new features that we developed. Our features included the cosine similarity of the word embeddings that we generated for each PPDB phrase (the **MVLSASim** score), as well as lexical overlap features, features derived from WordNet, and distributional similarity features. We weighted the contribution of these features using ridge regression with its regularization parameter tuned using cross validation on the training data.

We calculate the correlation of the different ways of automatically ranking the paraphrases against the 26k human judgments that we collected. Figure 8 plots the different automatic paraphrase scores against the 5-point human judgments for four different ways of ranking the paraphrases:

1) the original paraphrase probability defined by Bannard and Callison-Burch (2005),
2) the heuristic ranking that Ganitkevitch et al. (2013) defined for PPDB 1.0,
3) the cosine similarity of word2vec embeddings. For phrases, we use the vector of the rarest word as an approximation of the vector for the phrase.
4) the new score predicted by our discriminative model, recorded as the **PPDB 2.0 SCORE**.

The paraphrase probability has a Spearman correlation of 0.41. The heuristic PPDB 1.0 ranking has a similar correlation of ρ = 0.41. The word2vec similarity improves correlation slightly to 0.46. To test our supervised method, we use cross validation: in each fold, we hold out 200 phrases along with all of their associated paraphrases for testing. Our rankings for PPDB 2.0 dramatically improve correlation with human judgments to ρ = 0.71.

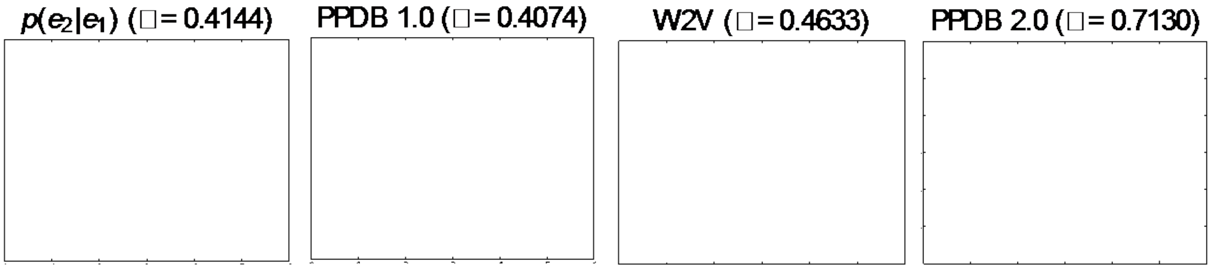


Figure 8: Scatterplots of automatic paraphrase scores (vertical axis) versus human scores (horizontal axis) for four ways of automatically ranking the paraphrases

As a qualitative example of the improvements to the paraphrase rankings, here are the top-10 ranked paraphrases under the PPDB 1.0 score for the input phrase *berries:*

1) embayments
2) strawberries
3) racks
4) grains
5) raspberries
6) blueberries
7) fruits
8) fruit
9) blackberries
10) beans

The top-10 ranked paraphrases under the PPDB 2.0 score for *berries* are:

1) strawberries
2) raspberries
3) blueberries
4) blackberries
5) fruits
6) fruit
7) beans
8) grains
9) seeds
10) kernels

3.5 Attaching a Semantic Entailment Relations to Paraphrases

We added an interpretable semantics to PPDB. Until we did so, the relationship between phrase pairs in the database has been weakly defined as approximately equivalent. We showed that paraphrase pairs in PPDB actually represent a variety of relations, including directed entailment (little girl/girl) and (rarely) pairs that represent antonyms or logical exclusion (nobody/someone). We automatically assign semantic entailment relations to entries in PPDB using features derived from past work on discovering inference rules from text and semantic taxonomy induction.

The selection of entailment relations that we used for PPDB was inspired by the relations from Bill MacCartney's thesis on natural language inference (MacCartney, 2009). He outlines 7 basic entailment relationships:

1) Equivalence ($P \equiv Q$): $\forall x[P(x) \leftrightarrow Q(x)]$

2) Forward Entailment ($P \sqsubset Q$): $\forall x[P(x) \rightarrow Q(x)]$
3) Reverse Entailment ($P \sqsupset Q$): $\forall x[Q(x) \rightarrow P(x)]$
4) Negation ($P \hat{} Q$): $\forall x \neg \forall x[P(x) \leftrightarrow Q(x)]$
5) Alternation ($P|Q$): $\forall x \neg[P(x) \wedge Q(x)]$
6) Cover ($P \smile Q$): $\forall x[P(x) \vee Q(x)]$
7) Independence ($P\#Q$): All other cases.

These relations are based on the theory of *natural logic*, meaning they are defined between pairs of natural language expressions rather than requiring an external formal representation. This made them an ideal fit for the phrase pairs in PPDB and similar automatically-constructed paraphrase resources.

Table 5: Column 1 gives the semantics of each label under MacCartney's Natural Logic. Column 2 gives the notation we use throughout the remainder of this paper. Column 3 gives a description that was shown to our annotators.

Natural logic	This project	Description
≡	≡	X is the same as Y
⊏	⊏	X is more specific than/is a type of Y
⊐	⊐	X is more general than/encompasses Y
ˆ |	¬	X is the opposite of Y X is mutually exclusive with Y
#	~ #	X is related in some other way to Y X is not related to Y

3.5.1 Annotation of semantic entailment types

We used Amazon Mechanical Turk (MTurk) to collect labels for our phrase pairs. We asked workers to choose between the options show in Table 5, which represent a modified version of MacCartney's relations. We replace negation (ˆ) with the weaker notion of "opposites," effectively merging it with the alternation (|) relation; we split the independent (#) class into two cases: truly independent phrases and phrases which are related by something other than entailment (which we denote ~). We omit the cover (‿) relation entirely, as its practicality is not obvious. We show each pair to 5 workers, taking the majority label as truth. Each HIT consisted of two control questions taken from WordNet. Workers achieved good accuracies on our controls (82% overall) and moderate levels of agreement (Fleiss's $\kappa = 0.56$) (Landis and Koch, 1977).

Based on our manual annotations we were able to estimate distributions over the semantic entailment types that occur in paraphrase pairs in PPDB. PPDB was released in six sizes (S, M, L, XL, XXL and XXXL), which fall roughly on a continuum from highest precision and lowest recall to lowest average precision and highest recall. Figure 9 shows how the distribution of entailment relations differs across the sizes of PPDB. PPDB-XXXL contains over 77MM paraphrase pairs (where the majority type is independent), compared to only 700K in PPDB-S (where the majority type is equivalent).

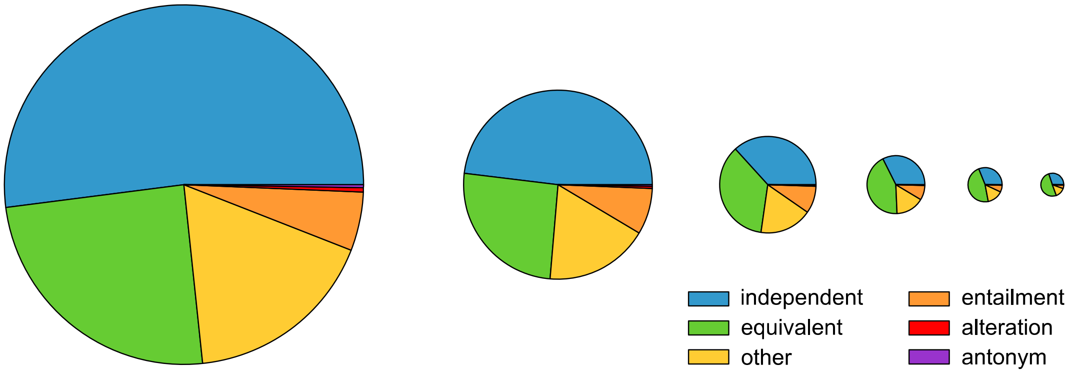


Figure 9: Distribution of entailment relations in different sizes of PPDB.

Our goal is to make these relations explicit, by providing annotations for each phrase pair. Because of the enormous scale of PPDB, this annotation must be done automatically.

3.5.2 Automatic classification of semantic entailment types

We built a classifier to automatically assign entailment types to entries in the PPDB, and demonstrated that it performs well both intrinsically and extrinsically. We fixed the direction of the and relations to create a single class and train a logistic regression classifier to distinguish between the 5 classes {#, ≡, ⊐, ¬, ~}. We computed a variety of basic lexical features and WordNet features (summarized in Table 6). We categorized the remaining features into two broad groups: monolingual features, which are based on observed usage in the Annotated Gigaword corpus (Napoles et al., 2012), and bilingual features, which are based on translation probabilities observed in bilingual parallel corpora.

Table 6: Top scoring pairs (x/y) according to various similarity measures, along with their manually classified entailment labels. Column 1 is cosine similarity based on dependency contexts. Column 2 is based on Lin (1998), column 3 on Weeds (2004), and column

Cosine Similarity		Monolingual (symmetric)		Monolingual (asymmetric)		Bilingual	
⊐	shades/the shade	¬	large/small	⊐	boy/little boy	≡	dad/father
⊐	yard/backyard	≡	few/several	⊐	man/two men	⊐	some kid/child
#	each other/man	¬	different/same	⊐	child/three children	≡	a lot of/many
⊐	picture/drawing	¬	other/same	≡	is playing/play	≡	female/woman
∼	practice/target	¬	put/take	⊐	side/both sides	≡	male/man

3.5.2.1 Monolingual features

Path features Snow et al. (2004) used lexico-syntactic patterns to mine taxonomic relations (hypernyms and hyponyms) between noun pairs. They were able to verify the earlier work of Hearst (1992) which found that certain patterns, e.g. *X and other Y*, are strong indicators of hypernymy.

Using similar path features, we learn new patterns to differentiate between more subtle relations. For example, we learn the pattern *separate X from Y* is highly indicative of the ¬ relation. We learn that the pattern *X including Y* suggests more than it suggests ≡ whereas the pattern *X known as Y* suggests ≡ more than ⊐. Table 7 gives examples of some of the paths most indicative of the ¬ relation.

Table 7: Top paths associated with the ¬ class

in X and in Y	*in foods and in beverages*
separate X from Y	*separate the old from the young*
to X and to Y	*to the left or to the right*
from X to Y	*from 7a.m. to 10p.m.*
more/less X than Y	*more harm than good*

3.5.3 Distributional features

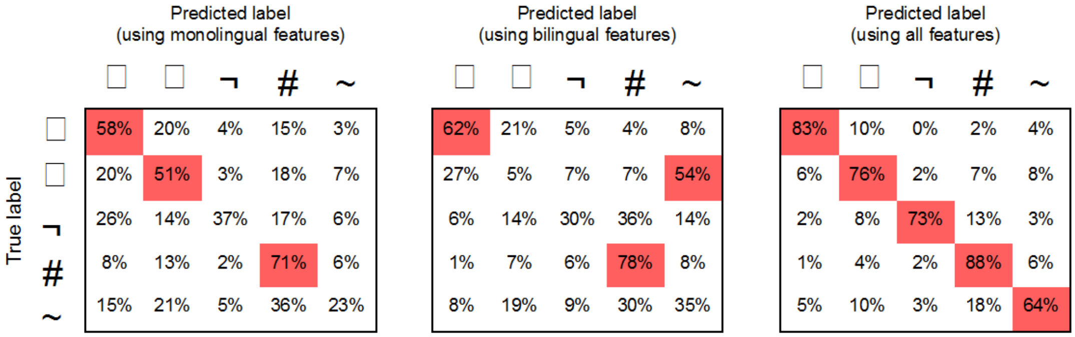
Lin and Pantel (2001) attempted to mine inference rules from text by finding paths in a dependency tree which connect the same nouns. The intuition is that good paraphrases should tend to modify and be modified by the same words. Given context vectors, Lin and Pantel (2001) used a symmetric similarity metric (Lin, 1998) to find candidate paraphrases. We build dependency context vectors for each word in our data and compute both symmetric as well as more recently proposed asymmetric similarity measures (Weeds et al., 2004; Szpektor and Dagan, 2008; Clarke, 2009), which are potentially better suited for identifying paraphrases. Table 6 gives a comparison of the pairs which are considered "most similar" according to several of these metrics.

3.5.4 Bilingual features

We explored a variety of bilingual features, which we expect to provide complimentary signals to the monolingual features. Each pair in PPDB is associated with several paraphrase probabilities, which are based on the probabilities of aligning each word to the foreign "pivot" phrase (a foreign translation shared by the two phrases), computed as described in Bannard and Callison-Burch (2005). We also compute the total number of shared foreign translations for each phrase pair. Table 6 shows the highest ranked pairs by this bilingual similarity score, in comparison to several of the monolingual scores.

3.5.5 Analysis

Table 7 shines some light onto the differences between monolingual and bilingual similarities. While the monolingual asymmetric metrics are good for identifying ⊐ pairs, the symmetric metrics consistently identify ¬ pairs; none of the monolingual scores we explored were effective in making the subtle distinction between ≡ pairs and the other types of paraphrases. In contrast, the bilingual similarity metric is fairly precise for identifying ≡ pairs, but provides less information for distinguishing between types of non-equivalent paraphrase. These differences are further exhibited in the confusion matrices shown in Figure 10; when the classifier is trained using only monolingual features, it misclassifies 26% of ¬ pairs as ≡, whereas the bilingual features make this error only 6% of the time. On the other hand, the bilingual features completely fail to predict the class, calling over 80% of such pairs ≡ or ~.



Predicted label (using monolingual features)

	□	□	¬	#	~
□	58%	20%	4%	15%	3%
□	20%	51%	3%	18%	7%
¬	26%	14%	37%	17%	6%
#	8%	13%	2%	71%	6%
~	15%	21%	5%	36%	23%

Predicted label (using bilingual features)

	□	□	¬	#	~
□	62%	21%	5%	4%	8%
□	27%	5%	7%	7%	54%
¬	6%	14%	30%	36%	14%
#	1%	7%	6%	78%	8%
~	8%	19%	9%	30%	35%

Predicted label (using all features)

	□	□	¬	#	~
□	83%	10%	0%	2%	4%
□	6%	76%	2%	7%	8%
¬	2%	8%	73%	13%	3%
#	1%	4%	2%	88%	6%
~	5%	10%	3%	18%	64%

Figure 10: Confusion matrices for classifier trained using only monolingual features (distributional and path) versus bilingual features (paraphrase and translation). True labels are shown along rows, predicted along columns. The matrix is normalized along rows, so that the predictions for each (true) class sum to 100%.

Our automatic classifier allows semantic entailment relations to be applied to a large-scale paraphrase resource like PPDB. The entailment relations given by natural logic are a great fit for paraphrase resources, since natural logic operates on pairs of natural language expressions (like the entries in PPDB). By classifying paraphrase entries with entailment relations, we provide them with an interpretable semantics. Our classifier uses extensive feature sets to scale natural logic to the enormous number of phrase pairs in PPDB. We evaluated our model, and demonstrated high accuracy on an intrinsic task. On an extrinsic RTE task, our model's predictions allow an RTE system to find 17% more proofs and achieve a higher overall accuracy than when using WordNet's manual relations.

3.5.6 Example entailment assignments

Here are examples of paraphrases and the entailment types that the classifier assigns to them:

[NN] | humankind | mankind | Equivalence
[JJ] | cross-disciplinary | interdisciplinary | Equivalence
[JJ] | anti-us | anti-american | Equivalence
[JJ] | eco-friendly | environmentally-friendly | Equivalence
[JJ] | crucial | vital | Equivalence
[VBN] | mentioned | cited | Equivalence
[VBG] | safeguarding | protecting | Equivalence
[NNS] | perspectives | viewpoints | Equivalence
[NNS] | astronauts | cosmonauts | Equivalence
[VBN] | overthrown | toppled | Equivalence

[VBD] | re-examined | examined | ForwardEntailment
[NNS] | t-shirts | shirts | ForwardEntailment
[NNS] | videotapes | tapes | ForwardEntailment
[VBD] | wrestled | struggled | ForwardEntailment
[NNS] | singers | performers | ForwardEntailment
[NNS] | embargoes | sanctions | ForwardEntailment
[NNS] | policemen | officers | ForwardEntailment
[NNS] | spirits | drinks | ForwardEntailment
[NNP] | shoppers | customer | ForwardEntailment
[NNS] | flecks | markings | ForwardEntailment

[NNS] | storms | rainstorms | ReverseEntailment
[NNS] | agreements | treaty | ReverseEntailment
[VBG] | fuelling | refuelling | ReverseEntailment
[NNS] | books | handbooks | ReverseEntailment
[NN] | area | region | ReverseEntailment
[NNS] | destruction | disasters | ReverseEntailment
[VBG] | bombing | firebombing | ReverseEntailment
[NN] | scarf | headscarf | ReverseEntailment
[NNS] | paths | footpaths | ReverseEntailment
[NNS] | applicants | claimants | ReverseEntailment

[NNS] | females | males | Exclusion
[VBN] | decreased | increased | Exclusion
[NNS] | debtors | creditors | Exclusion
[NNS] | children | mothers | Exclusion
[JJ] | illiterate | literate | Exclusion
[NNS] | whites | blacks | Exclusion
[JJ] | impossible | possible | Exclusion
[JJ] | downwind | upwind | Exclusion
[JJ] | lawful | unlawful | Exclusion
[JJ] | horizontal | vertical | Exclusion

[JJ] | paedophilia | paedophile | OtherRelated
[NN] | chaplain | chaplaincy | OtherRelated
[JJ] | nihilist | nihilistic | OtherRelated
[NN] | murderer | murder | OtherRelated
[NN] | constitution | constitutional | OtherRelated

3.6 Clustering Paraphrases by Word Sense

A primary benefit of PPDB is its enormous scale, and the fact that it has better coverage than manually compiled resources like WordNet (Miller, 1995). However, our automatically generated paraphrase resources had a drawback that it grouped all senses of polysemous words together, and did not partition paraphrases into groups like WordNet does with its synsets. Thus a search for paraphrases of the noun *bug* would yield a single list of paraphrases that includes *insect, glitch,* *beetle, error, microbe, wire, cockroach, malfunction, microphone, mosquito, virus, tracker, pest, informer, snitch, parasite, bacterium, fault, mistake, failure* and many others. Therefore, we designed algorithms to group our paraphrases into clusters that denote the distinct senses of the input word or phrase, as shown in Figure 4.

To create word sense clusters, we applied two clustering algorithms, Hierarchical Graph Factorization Clustering (Yu et al., 2005; Sun and Korhonen, 2011) and Self-Tuning Spectral Clustering (Ng et al., 2001; Zelnik-Manor and Perona, 2004), and systematically explore different ways of defining the similarity matrix that they use as input. Both of our clustering algorithms take as input an adjacency matrix W where the entries w_{ij} correspond to some measure of similarity between words i and j. W is a 20x20 matrix like shown in Figure 11 that specifies the similarity of every pair of paraphrases like *microbe* and *bacterium* or *microbe* and *malfunction*.

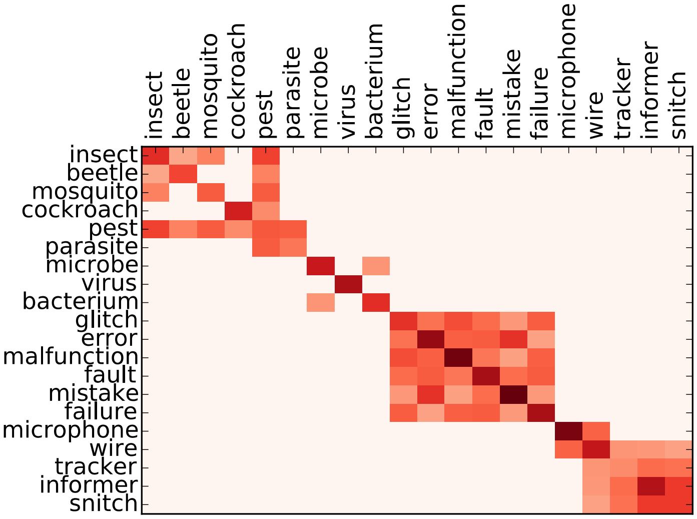


Figure 11: An affinity matrix showing the similarity between each pair of paraphrases for the word *bug*.

We systematically investigated four different types of similarity scores to populate *W*. We exploit a variety of features from PPDB to cluster its paraphrases by sense, including

1. its implicit graph structure,
2. aligned foreign words,
3. paraphrase scores,
4. monolingual distributional similarity scores.

Our goal was to determine which algorithm and features are the most effective for clustering paraphrases by sense. We address three research questions:

- Which similarity metric is best for sense clustering? We systematically compared different ways of defining matrices that specify the similarity between pairs of paraphrases.
- Are better clusters produced by comparing second-order paraphrases? We used PPDB's graph structure to decide whether *mosquito* and *pest* belong to the same sense cluster by comparing lists of paraphrases for the two words.

- Can entailment relations inform sense clustering? We used our model's predicted semantic entailments like *beetle* is-a *insect*, and its prediction that is no entailment between *malfunction* and *microbe*.

Our method produced sense clusters that are qualitatively and quantitatively good, and that represent a substantial improvement to the PPDB resource.

Our sense clustering work is closely related to the task of word sense induction (WSI), which aims to discover all senses of a target word from large corpora. One family of common approaches to WSI aims to discover the senses of a word by clustering the monolingual contexts in which it appears (Navigli, 2009). Another uncovers a word's senses by clustering its foreign alignments from parallel corpora (Diab, 2003). A more recent family of approaches to WSI represents a word as a feature vector of its substitutable words, i.e. paraphrases (Melamud et al., 2015; Yatbaz et al., 2012). Our algorithm took inspiration from each of these families of approaches, and we explored them when measuring word similarity in sense clustering.

The work most closely related to ours is that of Apidianaki et al. (2014), who used a simple graph-based approach to cluster pivot paraphrases on the basis of contextual similarity and shared foreign alignments. Their method represents paraphrases as nodes in a graph and connects each pair of words sharing one or more foreign alignments with an edge weighted by contextual similarity. Concretely, for paraphrase set *P*, it constructs a *graph* $G = (V, E)$ where vertices $V = \{p_i \in P\}$ are words in the paraphrase set and edges connect words that share foreign word alignments in a bilingual parallel corpus. The edges of the graph are weighted based on their contextual similarity (computed over a monolingual corpus). In order to partition the graph into clusters, edges in the initial graph *G* with contextual similarity below a threshold *T′* are deleted. The connected components in the resulting graph *G′* are taken as the sense clusters. The threshold is dynamically tuned using an iterative procedure (Apidianaki and He, 2010).

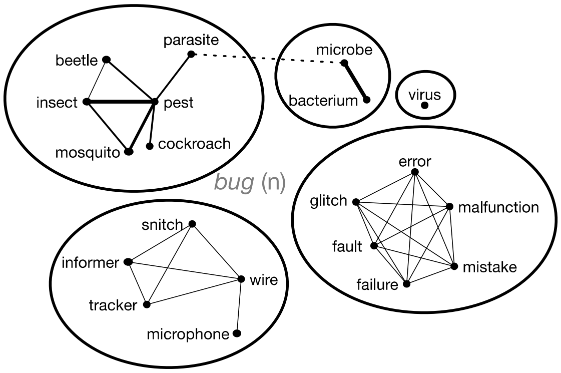


Figure 12: Apidianaki's algorithm connects all paraphrases that share foreign alignments, and cuts edges below a dynamically-tuned cutoff weight (dotted lines). The resulting connected components are its clusters.

3.6.1 Graph Clustering Algorithms

To partition paraphrases by sense, we used two advanced graph clustering methods rather than using Apidianaki et al. (2014)'s edge deletion approach. Both of them allowed us to experiment with a variety of similarity metrics.

3.6.1.1 Hierarchical Graph Factorization Clustering

The Hierarchical Graph Factorization Clustering (HGFC) method was developed by Yu et al. (2006) to probabilistically partition data into hierarchical clusters that gradually merge finer-grained clusters into coarser ones. Sun and Korhonen (2011) applied HGFC to the task of clustering verbs into Levin (1993)-style classes. Sun and Korhonen extended the basic HGFC algorithm to automatically discover the latent tree structure in their clustering solution and incorporate prior knowledge about semantic relationships between words. They showed that HGFC far outperformed agglomerative clustering methods on their verb data set. We adopted Sun and Korhonen's implementation of HGFC for our experiments.

HGFC takes as input a nonnegative, symmetric adjacency matrix $W = \{w_{ij}\}$ where rows and columns represent paraphrases $p_i \in P$, and entries w_{ij} denote the similarity between paraphrases *simD* (p_i, p_j). The algorithm works by factorizing W into a bipartite graph, where the nodes on one side represent paraphrases, and nodes on the other represent senses. The output of HGFC is a set of clusters of increasingly coarse granularity, which we can also represent with a tree structure. The algorithm automatically determines the number of clusters at each level. For our task, this has the benefit that a user can choose the cluster granularity most appropriate for the downstream. Another benefit of HGFC is that it probabilistically assigns each paraphrase to a cluster at each level of the hierarchy. If some p_i has high probability in multiple clusters, we can assign p_i to all of them.

3.6.1.2 Spectral Clustering

The second clustering algorithm that we use is Self-Tuning Spectral Clustering (Zelnik-Manor and Perona, 2004). Like HGFC, spectral clustering takes an adjacency matrix W as input, but the similarities end there. Whereas HGFC produces a hierarchical clustering, spectral clustering produces a flat clustering with k clusters, with k specified at runtime. The Zelnik-Manor and Perona (2004)'s self-tuning method is based on Ng et al. (2001)'s spectral clustering algorithm, which computes a normalized Laplacian matrix L from the input W, and executes K-means on the largest k eigenvectors of L. Intuitively, the largest k eigenvectors of L should align with the k senses in our paraphrase set.

3.6.2 Similarity Measures

Each of our clustering algorithms take as input an adjacency matrix W where the entries w_{ij} correspond to some measure of similarity between words i and j. For the paraphrases in Figure 4, W is a 20x20 matrix that specifies the similarity of every pair of paraphrases like *microbe* and *bacterium* or *microbe* and *malfunction*. We systematically investigated four types of similarity scores to populate W.

3.6.2.1 Paraphrase Scores

We used supervised logistic regression to combine a variety of scores so that they align with human judgements of paraphrase quality into the **PPDB 2.0 Score.** It is a nonnegative real number that can be used directly as a similarity measure.

$$w_{ij} = \begin{cases} PPDB\ 2.0\ Score(i,j) & (i,j) \in PPDB \\ 0 & otherwise \end{cases} \qquad \textbf{(10)}$$

3.6.2.2 Second-Order Paraphrase Scores

We defined two novel similarity metrics that calculate the similarity of words *i* and *j* by comparing their second-order paraphrases. Instead of comparing *microbe* and *bacterium* directly with their PPDB 2.0 score, we look up all of the paraphrases of *microbe* and all of the paraphrases of *bacterium*, and compare those.

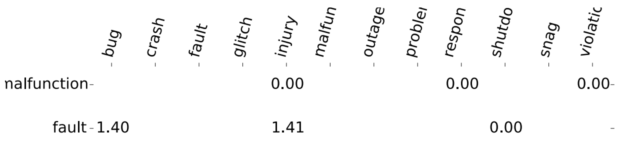


Figure 13: Comparing second-order paraphrases for malfunction and fault based on word-paraphrase vectors. The value of vector element v_{ij} is PPDB 2.0 Score(i, j).

Specifically, we form notional *word-paraphrase* feature vectors v_i^p and v_j^p where the features correspond to words with which each is connected in PPDB, and the value of the *k*th element of v_i^p equals PPDB 2.0 Score(i, k). We can then calculate the cosine similarity or Jensen-Shannon divergence between vectors:

$$sim_{PPDB_{cos}}(i,j) = \cos(v_i^p, v_j^p) \qquad \textbf{(11)}$$

$$sim_{PPDB_{js}}(i,j) = 1 - \mathrm{JS}(v_i^p, v_j^p) \qquad \textbf{(12)}$$

where $\mathrm{JS}(v_i^p, v_j^p)$ is calculated assuming that the paraphrase probability distribution for word *i* is given by its normalized *word-paraphrase* vector v_i^p.

3.6.2.3 Similarity of Foreign Word Alignments

When an English word is aligned to several foreign words, sometimes those different translations indicate a different word sense. Using this intuition, Gale et al. (1992) trained an English WSD system on a bilingual corpus, using the different French translations as labels for the English word senses. For instance, given the English word *duty*, the French translation *droit* was a proxy for its *tax* sense and *devoir* for its *obligation* sense.

PPDB is derived from bilingual corpora. We used the aligned foreign words and their associated translation probabilities that underlie each PPDB entry. For each English word in our dataset, we got each foreign word that it aligns to in the Spanish and Chinese bilingual parallel corpora. We use this to define a novel foreign word alignment similarity metric, $\mathrm{sim}_{\mathrm{TRANS}}(i,j)$ for two English paraphrases *i* and *j*. This is calculated as the cosine similarity of the *word-alignment* vectors v_i^a and v_j^a where each feature v^a is a foreign word to which *i* or *j* aligns, and the value of the entry v_{if}^a is the translation probability $p(f|i)$.

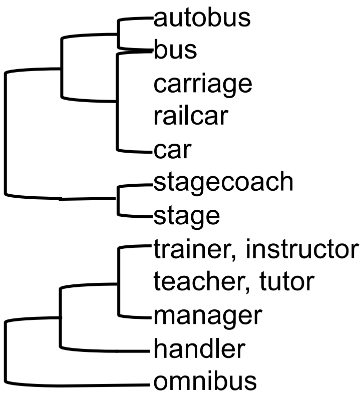
$$sim_{TRANS}(i,j) = \cos(v_i^a, v_j^a) \qquad \textbf{(13)}$$

3.6.2.4 Monolingual Distributional Similarity

Finally, we populated the adjacency with a distributional similarity measure based on word2vec (Mikolov et al., 2013). Each paraphrase i in our data set is represented as a 300-dimensional word2vec embedding v_{iw} trained on part of the Google News dataset. Phrasal paraphrases that did not have an entry in the word2vec dataset are represented as the mean of their individual word vectors. We use the cosine similarity between word2vec embeddings as our measure of distributional similarity.

3.6.3 Determining the Number of Senses

The optimal number of clusters for a set of paraphrases will vary depending on how many senses there ought to be for an input word like *bug*. It is generally recognized that optimal sense granularity depends on the application (Palmer et al., 2001). WordNet has notoriously fine-grained senses, whereas most word sense disambiguation systems achieve better performance when using coarse-grained sense inventories (Navigli, 2009). Depending on the task, the sense clustering for query word *coach* in Figure 14 with k = 5 clusters may be preferable to the alternative with k = 3 clusters. An ideal algorithm for our task would enable clustering at varying levels of granularity to support different downstream NLP applications.



(a) HGFC clustering result

k=5	c_1: trainer, tutor, instructor, teacher c_2: stagecoach, stage c_3: omnibus, bus, autobus c_4: car, carriage, railcar c_5: manager, handler
k=3	c_1: trainer, tutor, instructor, teacher, manager, handler c_2: stagecoach, stage c_3: omnibus, bus, autobus, car, carriage, railcar

(b) Spectral clustering results

Figure 14: Sense clusters for the word *coach*

Both of our clustering algorithms can produce sense clusters at varying granularities. For HGFC this requires choosing which level of the resulting tree structure to take as a clustering solution, and for spectral clustering we must specify the number of clusters prior to execution. To determine the optimal number of clusters, we use the mean Silhouette Coefficient (Rousseeuw, 1987) which balances optimal inter-cluster tightness and intra-cluster distance. The Silhouette Coefficient is calculated for each paraphrase p_i as

$$s(p_i) = \frac{b(p_i)-a(p_i)}{\max\{a(p_i),b(p_i)\}} \qquad \textbf{(14)}$$

where $a(p_i)$ is p_i's average intra-cluster distance (average distance from p_i to each other p_j in the same cluster), and $b(p_i)$ is p_i's lowest average inter-cluster distance (distance from p_i to the nearest external cluster centroid). For each clustering algorithm, we choose as the 'solution' the clustering which produces the highest mean Silhouette Coefficient. The Silhouette Coefficient calculation takes as input a matrix of pairwise distances, so we simply use $1 - W$ where the adjacency matrix W is calculated using one of the similarity methods we defined.

3.6.4 Incorporating Entailment Relations

We used the automatically predicted semantic entailment relations to refine the clustering. While a negative entailment relationship (*Exclusive* or *Independent*) does not preclude words from belonging to the same sense of some query word, a positive entailment relationship (*Equivalent*, *Forward/Reverse Entailment*) does give a strong indication that the words belong to the same sense.

We used a straightforward way to determine whether entailment relations provide information that is useful to the final clustering algorithm. Both of our algorithms take an adjacency matrix W as input, so we add entailment information by simply multiplying each pairwise entry by its entailment probability. Specifically, we set

$$w_{ij} = \begin{cases} (1 - p_{ind}(i,j))sim_D(i,j) & (i,j) \in PPDB \\ 0 & otherwise \end{cases} \qquad (15)$$

where $p_{ind}(i,j)$ gives the probability that there is an *Independent* entailment relationship between words i and j. Intuitively, this should increase the similarity of words that are very likely to be entailing like *fault* and *failure*, and decrease the similarity of non-entailing words like *cockroach* and *microphone*.

3.7 Domain Specific Paraphrases

We developed an algorithm to differentiate paraphrases by domain. As illustrated in Figure 15, paraphrases that are highly probable in the general domain (e.g. *hot* = *sexy*) can be extremely improbable in more specialized domains like biology. Dominant word senses change depending on domain: the verb *treat* is used in expressions like *treat you to dinner* in conversational domains versus *treat an infection* in biology. This domain shift changes the acceptability of its paraphrases.

	General	Biology
hot	warm, sexy, exciting	heated, warm, thermal
treat	address, handle, buy	cure, fight, kill
head	leader, boss, mind	skull, brain, cranium

Figure 15: Examples of domain-sensitive paraphrases. Most paraphrase extraction techniques learn paraphrases for a mix of senses that work well in general. But in specific domains, paraphrasing should be sensitive to specialized language use.

We addressed the problem of customizing paraphrase models to specific target domains. We explored the following ideas:

1. We sorted sentences in the training corpus based on how well they represent the target domain, and then extract paraphrases from a subsample of the most domain-like data.
2. We improved our domain-specific paraphrases by weighting each training example based on its domain score, instead of treating each example equally.
3. We improved recall while maintaining precision by combining the subsampled in-domain paraphrase scores with the general-domain paraphrase scores.

3.7.1 Sorting by domain specificity

The crux of our method is to train a paraphrase model on data from the same domain as the one in which the paraphrases will be used. In practice, it is unrealistic that we will be able to find bilingual parallel corpora precompiled for each domain of interest. We instead subsample from a large bitext, biasing the sample towards the target domain.

We adapt and extend a method developed by Moore and Lewis (2010) (henceforth M-L), which builds a domain-specific sub-corpus from a large, general-domain corpus. The M-L method assigns a score to each sentence in the large corpus based on two language models, one trained on a sample of target domain text and one trained on the general domain. We want to identify sentences which are similar to our target domain and dissimilar from the general domain. M-L captures this notion using the difference in the cross-entropies according to each language model (LM). That is, for a sentence s_i, we compute

$$\sigma_i = H_{target}(s_i) - H_{general}(s_i) \qquad \textbf{(16)}$$

where H_{target} is the cross-entropy under the in-domain language model and $H_{general}$ is the cross-entropy under the general domain LM. Cross-entropy is monotonically equivalent to LM perplexity, in which lower scores imply a better fit. Lower σ_i signifies greater domain-specificity.

3.7.2 Domain-Specific Paraphrases

To apply the M-L method to paraphrasing, we need a sample of in-domain monolingual text. This data is not directly used to extract paraphrases, but instead to train an n-gram LM for the target domain. We compute σ_i for the English side of every sentence pair in our bilingual data, using the target domain LM and the general domain LM. We sort the entire bilingual training corpus so that the closer a sentence pair is to the top of the list, the more specific it is to our target domain.

We investigated several uses of the M-L algorithm:

1. We choose a threshold value for $\sigma_i$and discarding all sentence pairs that fall outside of that threshold, we can extract paraphrases from a subsampled bitext that approximates the target domain.
2. We tried weighting each training example proportional to σ_i when computing the paraphrase scores, instead of extracting from a subsampled corpus where each training example is equally weighted.
3. We combined multiple paraphrase scores: one derived from the original corpus and one from the subsample. This had the advantage of producing the full set of paraphrases that can be extracted from the entire bitext.

3.8 Paraphrase Databases for Other Languages

We released an expansion of the paraphrase database (PPDB) that includes a collection of paraphrases in 23 different languages. The resource is derived from large volumes of bilingual parallel data. The multilingual PPDB has over a billion paraphrase pairs in total, covering the following languages: Arabic, Bulgarian, Chinese, Czech, Dutch, Estonian, Finnish, French, German, Greek, Hungarian, Italian, Latvian, Lithuanian, Polish, Portuguese, Romanian, Russian, Slovak, Slovenian, and Swedish.

We used the pivoting technique described in Section 3.1 to extract foreign paraphrases in a similar fashion that we did to extract English paraphrases. Rather than pivoting over foreign language phrases to find related English expressions, we pivot over English to find pairs of foreign phrases that are paraphrases. Two expressions in language *F*, *f1* and *f2*, that translate to a shared expression *e* in another language *E* can be assumed to have the same meaning. We can thus find paraphrases of a German phrase like *inhaftiert* by pivoting over a shared English translation like *imprisoned* and extract German paraphrase pair ⟨*inhaftiert, verhaftet*⟩, as illustrated in Figure 16.

Since *inhaftiert* can have many possible translations, and since each of those can map back to many possible German phrases, we extract not only *verhaftet* as a paraphrase, but also *eingesperrt, festgenommen, eingekerkert, festgehalten, festnahmen, festnahme, statt, stattfinden, gefangenen, gefangengenommen, haft, innerhalb,* and others.

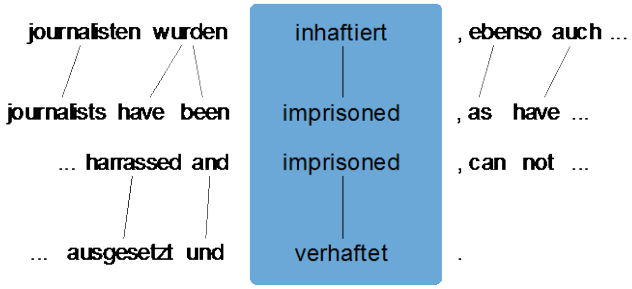


Figure 16: German paraphrases are extracted by pivoting over a shared English translation.

Paraphrases need not be extracted from a single pivot language. They can be obtained from multiple bitexts where the language of interest is contained on one side of the parallel corpus. Thus, instead of extracting German paraphrases just by pivoting over English, we could extract additional paraphrases from a German-French or a German-Spanish bitext. Although it is easy to construct parallel corpora for all pairs of languages in the European Union using existing resources like the Europarl parallel corpus (Koehn, 2005) or the JRC corpus (Steinberger et al., 2006), we only pivot over English for this release of the multilingual PPDB.

The reason that we limit ourselves to pivoting over English, is that we extend the bilingual pivoting method to incorporate syntactic information. Abundant NLP resources, such as statistical parsers, are available for English. By using annotations from the English side of the bitext, we are able to create syntactic paraphrases for languages for which we do not have syntactic parsers. We project the English syntax onto the foreign sentence via the automatic word alignments. The notion of projecting syntax across aligned bitexts has been explored for bootstrapping parsers (Hwa et al., 2005). Only the English side of each parallel corpus needs to be parsed, which we do with the

Berkeley Parser (Petrov et al., 2006). Figure 16 shows how a phrasal paraphrase can be generalized into a syntactic paraphrase by replacing words and phrases that are themselves paraphrases with appropriate nonterminal symbols.

NP → zwölf dieser Teilnehmer | 12 of the participants

NP → 12 die Beteiligten | 12 of the participants

combine to the phrasal paraphrase

NP → zwölf dieser Teilnehmer | 12 die Beteiligten

Similarly,

NP → CD dieser NNS | CD of the NNS

NP → CD die NNS | CD of the NNS

combine to the syntactic paraphrase

NP → CD dieser NNS | CD die NNS

Figure 17: We extract syntactic paraphrases for the foreign PPDBs. The syntactic labels are drawn from parse trees of the English sentences in our bitexts.

3.8.1 Resource Size

We extracted significantly different numbers of paraphrases for each of the languages. The number of paraphrases is roughly proportional to the size of the bitext that was used to extract the paraphrases for that language. Unsurprisingly, we observe a large difference in size between the French, Arabic, and Chinese paraphrase sets, and the others. This is due to the comparatively large bilingual corpora that we used for the three languages, versus the smaller bitexts that we used for the other languages (see Table 9). Table 8 gives a detailed breakdown of the number of each kind of paraphrases (lexical, phrasal, syntactic) that we have extracted for each language.

Table 8: An overview over the sizes of the multilingual PPDB. The number of extracted paraphrases varies by language, depending on the amount of data available as well as the languages morphological richness. The language names are coded following ISO 639-2.

Language	Code	Number of Paraphrases			
		Lexical	Phrasal	Syntactic	Total
Arabic	Ara	119.7M	45.1M	20.1M	185.7M
Bulgarian	Bul	1.3M	1.4M	1.2M	3.9M
Czech	Ces	7.3M	2.7M	2.6	12.1M
German	Deu	7.9M	15.4M	4.9M	28.3M
Greek	Ell	5.4M	9.4M	7.4M	22.3M
Estonian	Est	7.9M	1.0M	0.4M	9.2M
Finnish	Fin	41.4M	4.9M	2.3M	48.6M
French	Fra	78.8M	254.2M	170.5M	503.5M
Hungarian	Hun	3.8M	1.3M	0.2M	5.3M
Italian	Ita	8.2M	17.9M	9.7M	35.8M
Lithuanian	Lit	8.7M	1.5M	0.8M	11.0M
Latvian	Lav	5.5M	1.4M	1.0M	7.9M
Dutch	Nld	6.1M	15.3M	4.5M	25.9M
Polish	Pol	6.5M	2.2M	1.4M	10.1M
Portuguese	Por	7.0M	17.0M	9.0M	33.0M
Romanian	Ron	1.5M	1.8M	1.1M	4.5M
Russian	Rus	81M	46M	16M	144.4M
Slovak	Slk	4.8M	1.8M	1.7M	8.2M
Slovenian	Slv	3.6M	1.6M	1.4M	6.7M
Swedish	Swe	6.2M	10.3M	10.3M	26.8M
Chinese	Zho	52.5M	46.0M	8.9M	107.4M

Table 9: The sizes of the bilingual training data used to extract each language-specific version of PPDB.

Language	Sentence Pairs	Foreign Words	English Words	Corpora
Arabic	9,542,054	205,508,319	204,862,233	GALE
Bulgarian	406,934	9,306,037	9,886,401	Europarl-v7
Chinese	11,097,351	229,364,807	244,690,254	GALE
Czech	596,189	12,285,430	14,277,300	Europarl-v7
Dutch	1,997,775	49,533,217	50,661,711	Europarl-v7
Estonian	651,746	11,214,489	15,685,939	Europarl-v7
Finnish	1,924,942	32,330,289	47,526,505	Europarl-v7
French	52,004,519	932,475,412	821,546,279	Europarl-v7, 10^9 word parallel corpus, JRC, OpenSubtitles, UN
German	1,720,573	39,301,114	41,212,173	Europarl-v7
Greek	1,235,976	32,031,068	31,939,677	Europarl-v7
Hungarian	624,934	12,422,462	15,096,547	Europarl-v7
Italian	1,909,115	48,011,261	49,732,033	Europarl-v7
Latvian	637,599	11,957,078	15,412,186	Europarl-v7
Lithuanian	635,146	11,394,858	15,342,163	Europarl-v7
Polish	632,565	12,815,795	15,269,016	Europarl-v7
Portugese	1,960,407	49,961,396	49,283,373	Europarl-v7
Romanian	399,375	9,628,356	9,710,439	Europarl-v7
Russian	2,376,138	40,765,979	43,273,593	CommonCrawl, Yandex 1M corpus, News Commentary
Slovak	640,715	15,442,442	12,942,700	Europarl-v7
Slovenian	623,490	12,525,860	15,021,689	Europarl-v7
Swedish	1,862,234	45,767,032	41,602,279	Europarl-v7

3.8.2 Morphological Variants as Paraphrases

Many of the languages covered by our resource are more morphologically complex than English. Since we are using English pivot phrases and English syntactic labels, the pivoting approach tends to group a variety of morphological variants of a foreign word into the same paraphrase cluster. For example, French adjectives inflect for gender and number, but English adjectives do not. Therefore, the French words *grand*, *grande*, *grands* and *grandes* would all share the English translation *tall*, and would therefore all be grouped together as paraphrases of each other. It is unclear whether this grouping is desirable or not, and the answer may depend on the downstream task. It is clear that there are distinctions that are made in the French language that our paraphrasing method currently does not make.

This is also observable in verbs. Other languages often have more inflectional variation than English does. Whereas English verbs only distinguish between past versus present tense and 3rd person singular versus non-3rd person singular, other languages exhibit more forms. For instance, the English verb *go*, aligns to a variety of present forms of the French *aller*. The high-ranking paraphrases of *vais*, the first person singular form of *aller*, are all other forms of the verb. These are shown in Table 10. Similar effects can be observed across other verb paraphrases, both in French and other languages. The minimal distinction in the Penn Treebank tags between past tense verbs (VBD), base form verbs (VB) and present tense verbs (VBN/VBP), partitions the foreign verbs to some extent. But clearly there is a semantic distinction between verb forms that are marked for person and number, which our method is not currently making.

The interaction between out bilingual pivoting method and English's impoverished morphologic system, open up avenues for improving the quality of the multilingual paraphrases. Our method makes distinctions between paraphrases when they have different syntactic labels. This does a good job of separating out things that make a sense distinction based on part of speech (like *squash* which paraphrases as *racquetball* as a noun and *crush* as a verb). It also limits different paraphrases based on which form the original phrase takes. For instance, *divide* can paraphrase as *fracture* or *split* in both noun and verb forms, but it can only paraphrase as *gap* when the original phrase is a noun. We use Penn Treebank tags, which are rather English-centric. This tag set could be replaced or refined to make finer-grained distinctions that are present in the foreign language. Refined, language-specific tag sets would do a better job at partitioning paraphrase sets that should be distinct.

Table 10: Top paraphrases extracted for forms of the French *aller* and the German *denken*.

Tag	Phrase	Paraphrases
VB	vais	va, vas, irai, vont, allons, ira, allez, irons
	vas	va, vont, allez, vais, allons, aller
	vont	vas, va, allons, allez, vais, aller
VBD	allais	allait, alliez, allaient, allions
VB	denke	denken, denkt

Table 10 shows that the English POS label preserves the unifying morphological characteristic quite well: present tense forms of *aller* dominate the ranking for the VB (which best corresponds with present tense usage in English). Similarly, imperfect forms are reliably captured for the past tense VBD tag.

4 RESULTS AND DISCUSSIONS

4.1 Performing Natural Language Inference with Paraphrases

We evaluated the usefulness of the semantic entailments that we attached to PPDB by using them in a downstream task of recognizing textual entailment (RTE). We ran our experiments using Nutcracker, a state-of-the-art RTE system based on formal semantics (Bjerva et al., 2014). In the SemEval 2014 RTE challenge, this system performed in the top 5 out of the more than 20 participating systems (Marelli et al., 2014). Given a text/hypothesis (T/H) pair, Nutcracker uses the Boxer parser (Bos, 2008) to produce a formal semantic representation of both T and H, which it translates into standard first-order logic. The logical formulae are passed to an off-the-shelf theorem prover, which searches for a logical entailment, and to a model builder, which attempts to find a logical contradiction. By default, when the system fails to find a proof for either entailment or inconsistency, it predicts the most frequent class (in our case, NEUTRAL). Therefore, Nutcracker relies heavily on lexical entailment resources in order to improve the recall of the theorem prover and model builder.

4.1.1 Baselines

The most frequent class baseline is achieved by labeling every sentence pair as NEUTRAL, and results in an accuracy of 56%. A stronger baseline is obtained by running Nutcracker alone, without any external axioms; in this case, words are only equivalent if they are lemma-identical.

As an additional baseline, we generated a "basic" PPDB-XL knowledge base (KB), which consists exclusively of axioms expressing synonym relationships. I.e. for every pair of phrases ⟨p1, p2⟩ in PPDB-XL, the PPDB-XL KB contains the equivalence axiom syn(p1, p2). We also generated the WordNet KB, which is the default used by Nutcracker. This KB consists of axioms for all synonyms, antonyms, and hypernyms in WordNet, which generate *syn*, *isa*, and *isnota* axioms, respectively.

4.1.2 PPDB+

We converted our classifier's predictions into a set of axioms for Nutcracker. When our classifier predicted ≡ we generated an *syn* axiom, when it predicted ⊐ we generated an *isa* axiom, and when it predicted ¬ we generated an *isnota* axiom. # and ~ did not generate any axioms. We refer to this set of axioms as PPDB+. To calibrate our improvements, we also generated a KB using the human labels collected from MTurk, which we refer to as PPDB-Human.

4.2 Results

Table 11 shows Nutcracker's overall prediction accuracy and the number of proofs found when using each of the described KBs. Figure 8 shows the performance in terms of the precision and recall achieved for each of the three entailment classes: ENTAILMENT, CONTRADICTION, and NEUTRAL.

Table 11: Nutcracker's overall system accuracy and proof coverage when using different sources of axioms.

	Acc.	# Proofs	Coverage
MFC	56.4	0	0%
NC alone	74.3	878	17.8%
+ WN	77.5	1,051	21.3%
+ PPDB-XL	77.5	1,091	22.1%
+ PPDB+	78.0	1,197	24.3%
+ WN, PPDB+	**78.4**	**1,230**	**25.0%**
+ *WN, PPDB-H*	*78.6*	*1,232*	*25.0%*

Coverage is measured as the percent of sentence pairs for which NC's theorem prover or model builder is able to find a complete logical proof of either entailment or contradiction. When NC fails to find either type of proof, it guesses the most frequent class, NEUTRAL. NC alone uses no axioms. PPDB+ refers to the axioms generated automatically using the classifier described in this report. PPDB-H refers axioms generated using the human labels on which the classifier was trained.

Table 12 provides some examples of T/H pairs on which predictions differed using the PPDB+ compared to the WordNet KB.

Table 12: Examples of T/H pairs for which the system's prediction differed when using PPDB+ vs. WordNet

True	PPDB+	WN	Text/Hypothesis pair
Entail.	Entail.	Neutral	A **bride** in a white dress is running/A **girl** in a white dress is running.
Entail.	Neutral	Entail.	A **lemur** is biting a person's finger./An **animal** is biting a person's finger.
Contra.	Contra.	Neutral	**Someone** is playing a piano./There is **no one** playing a piano.
Contra.	Neutral	Contra.	There is **no man** pouring oil into a **pan**./**A man** is pouring oil into a **skillet**.

.

The PPDB+ KB outperforms all of the baselines, including the WordNet baseline, in overall accuracy as well as in F1 measure on all three entailment classes. The best results are achieved by combining the PPDB+ and WordNet KBs, reaching 78.4% overall accuracy. This improvement is due predominantly to increased recall; PPDB+ achieves 51% recall on the ENTAILMENT class, compared to only 44% when using WordNet, leading to a 5 point increase in F1 measure.

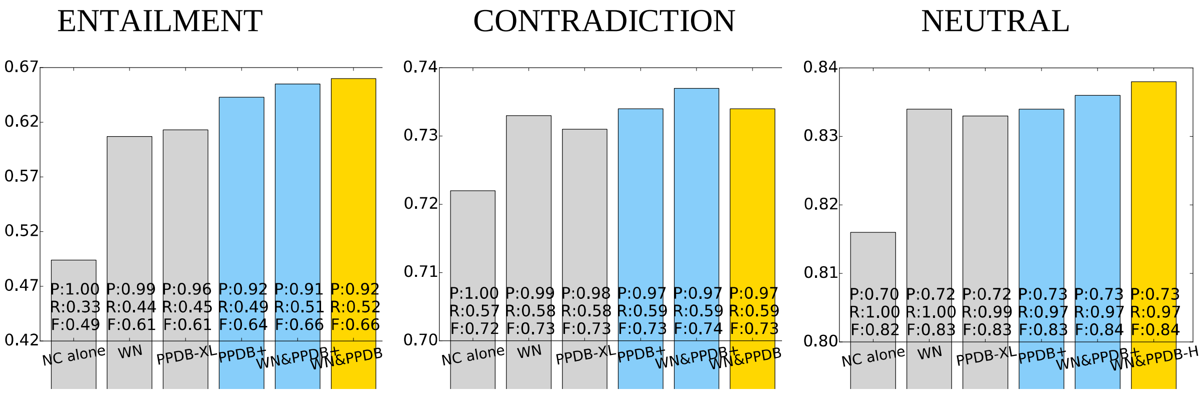


Figure 18: F1 measures achieved by Nutcracker on SICK test data when using various KBs.

Figure 18 shows the F1 measures for our various experiments with Nutcracker. Baselines are in gray, this work in blue, human references in gold. PPDB-XL refers to a run in which every pair which appears in PPDB is assumed to be equivalent. PPDB-H refers to a run in which manual labels were used to generate axioms. PPDB+ refers to runs in which the automatic classifications were used to generate axioms. In some cases, better proof coverage causes NC to find incorrect proofs, illustrated by the decreased performance on CONTRADICTION when using PPDB-H. For example, using PPDB-H, NC finds an inconsistency for the pair *Someone is not playing piano./A person is playing a keyboard*. Using the PPDB+, in which *piano/keyboard* is falsely classified as #, NC fails to find a proof and so correctly guesses NEUTRAL.

The improved recall is further illustrated by looking at the number of proofs that Nutcracker is able to find when using each of the KBs (Table 11). Recall that Nutcracker's entailment engine works by using a theorem prover to search for a logical entailment and a model builder to search for a logical inconsistency. When neither component succeeds in finding a proof, Nutcracker guesses NEUTRAL. Good proof coverage is therefore essential to Nutcracker's performance. PPDB+ enables Nutcracker to find a logical proof for 17% more sentence pairs relative to using WordNet only, providing an additional point in overall accuracy. Using PPDB+ Nutcracker achieves the same accuracy as it does when using PPDB-Human, demonstrating that the automatically generated PPDB+ provides as much utility to the end-to-end system as does a gold-standard resource.

4.3 Clustering Quality

We performed an intrinsic evaluation on our word-sense clustered paraphrases by comparing them against two gold standards.

4.3.1 Gold Standard Clusters

WordNet synsets provide a well-established basis for comparison, but only allow us to evaluate our method on the 38% of our PPDB dataset that overlaps it. We therefore evaluate performance on two test sets.

4.3.1.1 WordNet+

Our first test set is designed to assess how well our solution clusters align with WordNet synsets. We chose 185 polysemous words from the SEMEVAL 2007 dataset and an additional 16 hand-picked polysemous words. For each we formed a paraphrase set that was the intersection of their PPDB 2.0 XXXL paraphrases with their WordNet synsets, and their immediate hyponyms and hypernyms. Each reference cluster consisted of a WordNet synset, plus the hypernyms and hyponyms of words in that synset. On average, there are 7.2 reference clusters per paraphrase set.

4.3.1.2 CrowdClusters

Because the coverage of WordNet is small compared to PPDB, and because WordNet synsets are very fine-grained, we wanted to create a dataset that would test the performance of our clustering algorithm against large, noisy paraphrase sets and coarse clusters. For this purpose we randomly selected 80 query words from the SEMEVAL 2007 dataset and retrieved paraphrase sets from PPDB 2.0. We then iteratively organized each paraphrase set into reference senses with the help of crowd workers on Amazon Mechanical Turk. On average, there are 4.0 reference clusters per paraphrase set. A full description of our method is included in the supplemental materials.

4.3.2 Evaluation Metrics

We evaluated our method using two standard metrics: the paired F-Score and V-Measure. Both were used in the 2010 SemEval Word Sense Induction Task (Manandhar et al., 2010) and by Apidianaki et al. (2014). We give our results in terms of weighted average performance on these metrics, where the score for each individual paraphrase set is weighted by the number of reference clusters for that query word.

4.3.2.1 Paired F-Score

Paired F-Score frames the clustering problem as a classification task (Manandhar et al., 2010). It generates the set of all word pairs belonging to the same reference cluster, F(S), and the set of all word pairs belonging to the same automatically-generated cluster, F(K). Precision, recall, and F-score can then be calculated in the usual way, i.e.

$$Precision = \frac{F(K) \cap F(S)}{F(K)} \qquad \textbf{(17)}$$

$$Recall = \frac{F(K) \cap F(S)}{F(S)} \qquad \textbf{(18)}$$

$$F\text{–}Score = \frac{2 * Precision * Recall}{Precision + Recall} \qquad \textbf{(19)}$$

4.3.2.2 V-Measure

V-Measure assesses the quality of a clustering solution against reference clusters in terms of clustering homogeneity and completeness (Rosenberg and Hirschberg, 2007). Homogeneity describes the extent to which each cluster is composed of paraphrases belonging to the same reference cluster, and completeness refers to the extent to which points in a reference cluster are assigned to a single cluster. Both are defined in terms of conditional entropy. V-Measure is the harmonic mean of homogeneity and completeness;

$$V\text{–} Measure = \frac{2 * homogeneity * completeness}{homogeneity + completeness} \quad \textbf{(20)}$$

4.3.3 Baselines

We evaluate the performance of HGFC on each dataset against the following baselines:

4.3.3.1 Most Frequent Sense (MFS)

MFS assigns all paraphrases $p_i \in P$ to a single cluster. By definition, the completeness of the MFS clustering is 1.

4.3.3.2 One Cluster per Paraphrase (1C1PAR)

1C1PAR assigns each paraphrase $p_i \in P$ to its own cluster. By definition, the homogeneity of 1C1PAR clustering is 1.

4.3.3.3 Random (RAND)

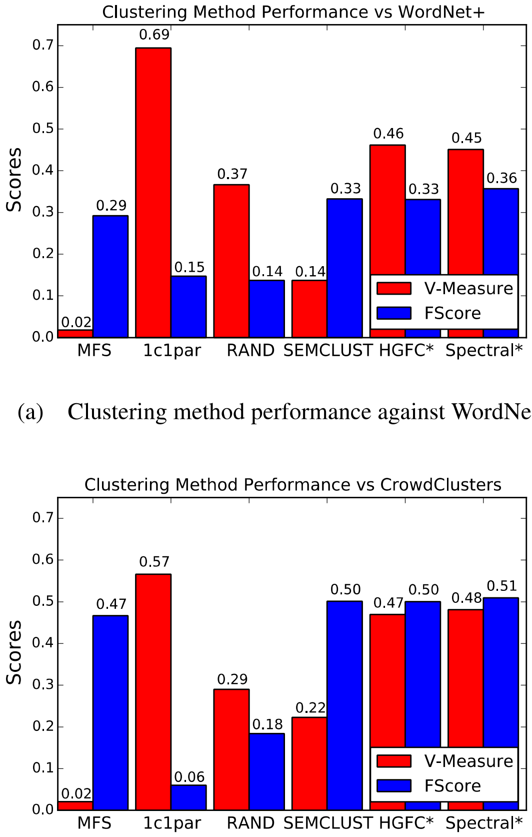
For each query term’s paraphrase set, we generate five random clusterings of k = 5 clusters. We then take F-Score and V-Measure as the average of each metric calculated over the five random clusterings.

4.3.3.4 SEMCLUST

We implement the SEMCLUST algorithm (Apidianaki et al., 2014) as a state-of-the-art baseline. Since PPDB contains only pairs of words that share a foreign word alignment, in our implementation we connect paraphrase words with an edge if the pair appears in PPDB. We adopt the WORD2VEC distributional similarity score for our edge weights.

4.3.4 Experimental Results

Figure 19 shows the performance of the two advanced clustering algorithms against the baselines. Our best configurations for HGFC and Spectral outperformed all baselines except 1C1PAR V-Measure, which is biased toward solutions with many small clusters (Manandhar et al., 2010), and performed only marginally better than SEMCLUST in terms of F-Score alone. The dominance of 1C1PAR V-Measure is greater for the WordNet+ dataset which has smaller reference clusters than CrowdClusters. Qualitatively, we find that methods that strike a balance between high F-Score and high V-Measure tend to produce the best clusters by human judgement. If we consider the average of F-Score and V-Measure as a comprehensive performance measure, our methods outperform all baselines.



(a) Clustering method performance against WordNet+

(b) Clustering method performance against CrowdClusters

Figure 19: Hierarchical Graph Factorization Clustering and Spectral Clustering both significantly outperform all baselines except 1C1PAR V-Measure.

On our dataset, the state-of-the-art SEMCLUST baseline tended to lump many senses of the query word together, and produced scores lower than in the original work. We attribute this to the fact that the original work extracted paraphrases from EuroParl, which is much smaller than PPDB, and thus created adjacency matrices W which were sparser than those produced by our method. Directly applied, SEMCLUST works well on small data sets, but does not scale well to the larger, noisier PPDB data. More advanced graph-based clustering methods produce better sense clusters for PPDB.

The first question we sought to address with this work was which similarity metric is the best for sense clustering. Figure 20 reports the average F-Score and V-Measure across 40 test configurations for each similarity calculation method. On average across test sets and clustering algorithms, the paraphrase similarity score (PPDB2.0Score) performs better than monolingual distributional similarity (simDISTRIB) in terms of F-Score, but the results are reversed for V-Measure. This is also shown in the best HGFC and Spectral configurations, where the two similarity scores are swapped between them.

Next, we investigated whether comparing second-order paraphrases would produce better clusters than simply using PPDB2.0Score directly. Table 13 also compares the two methods that we had for computing the similarity of second order paraphrases – cosine similarity (simPPDB.cos) and Jensen-Shannon divergence (simPPDB.JS). On average across test sets and clustering algorithms, using the direct paraphrase score gives stronger V-Measure and F-score than the second-order methods. It also produces coarser clusters than the second-order PPDB similarity methods.

Table 13: Average performance and number of clusters produced by our different similarity methods.

			Avg #
Method	**F-Score**	**V-Measure**	**Clusters**
$PPDB_{2.0}Score$	0.410	0.437	5.960
$sim_{DISTRIB}$	0.376	0.440	5.707
$sim_{PPDB.cos}$	0.389	0.428	7.204
$sim_{PPDB.JS}$	0.385	0.425	7.143
sim_{TRANS}	0.358	0.375	6.247
SEMCLUST	0.417	0.180	2.279
Reference	1.0	1.0	5.611

Finally, we investigated whether incorporating automatically predicted entailment relations would improve cluster quality, and we found that it did. All other things being equal, adding entailment information increases F-Score by .014 and V-Measure by .020 on average (Figure 20). Adding entailment information had the greatest improvement to HGFC methods with simDISTRIB similarities, where it improved F-Score by an average of .03 and V-Measure by an average of .05.

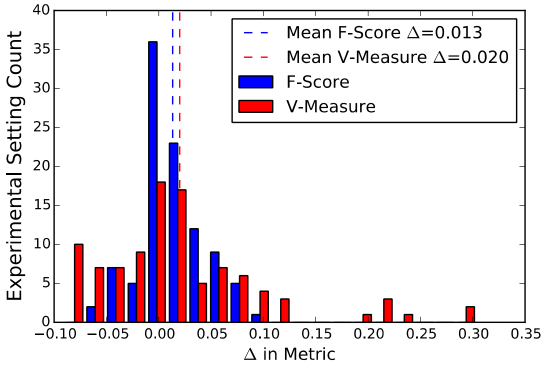
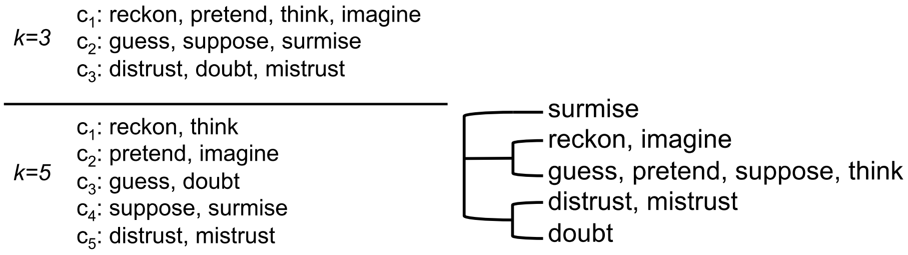


Figure 20: Histogram of metric change by adding entailment information across all experiments.

Figure 21 shows a qualitative example of the cluster produced by HPFC and Spectral Clustering for a polysemous verb in our test set.



k=3
c_1: reckon, pretend, think, imagine
c_2: guess, suppose, surmise
c_3: distrust, doubt, mistrust
k=5
c_1: reckon, think
c_2: pretend, imagine
c_3: guess, doubt
c_4: suppose, surmise
c_5: distrust, mistrust
surmise
reckon, imagine
guess, pretend, suppose, think
distrust, mistrust
doubt

(a) Spectral clustering results for *suspect* (v)

(b) HGFC clustering results for *suspect* (v)

Figure 21: Clusters produced our HGFC and Spectral Clustering methods for the verb *suspect*

4.4 Expanding Manually Created Lexical Semantic Resources

We increase the lexical coverage of FrameNet through automatic paraphrasing. We use crowdsourcing to manually filter out bad paraphrases in order to ensure a high-precision resource. Our expanded FrameNet contains an additional 22K lexical units, a 3-fold increase over the current FrameNet, and achieves 40% better coverage when evaluated in a practical setting on New York Times data.

Frame semantics describes a word in relation to real-world events, entities, and activities. Frame semantic analysis can improve natural language understanding (Fillmore and Baker, 2001), and has been applied to tasks like question answering (Shen and Lapata, 2007) and recognizing textual entailment (Burchardt and Frank, 2006; Aharon et al., 2010). FrameNet (Fillmore, 1982; Baker et al., 1998) is a widely-used lexical-semantic resource embodying frame semantics. It contains close to 1,000 manually defined *frames*, i.e. representations of concepts and their semantic properties, covering a wide array of concepts from Expensiveness to Obviousness.

Frames in FrameNet are characterized by a set of semantic roles and a set of lexical units (LUs), which are word/POS pairs that "evoke" the frame. For example, the following sentence contains a mention (i.e. *target*) of the Obviousness frame: *In late July, it was barely visible to the unaided eye*. This particular target instantiates several semantic roles of the Obviousness frame, including a Phenomenon (*it*) and a Perceiver (*the unaided eye*). Here, the LU *visible.a* evokes the frame. In total, the Obviousness frame has 13 LUs including *clarity.n, obvious.a,* and *show.v*.

The semantic information in FrameNet is broadly useful for problems such as entailment (Ellsworth and Janin, 2007; Aharon et al., 2010) and knowledge base population (Mohit and Narayanan, 2003; Christensen et al., 2010; Gregory et al., 2011), and is of general enough interest to language understanding that substantial effort has focused on building parsers to map natural language onto FrameNet frames (Gildea and Jurafsky, 2002; Das and Smith, 2012). In practice, however, FrameNet's usefulness is limited by its size. FN was built entirely manually by linguistic experts. As a result, despite many years of work, most of the words that one confronts in naturally occurring text do not appear at all in FN. For example, the word *blatant* is likely to evoke the Obviousness frame, but is not present in FN's list of LUs (Table 14). In fact, out of the targets we sample in this work (described in Section 4), fewer than 50% could be mapped to a correct frame using the LUs in FrameNet. This finding is consistent with what has been reported by Palmer and Sporleder (2010). Such low lexical coverage prevents FN from applying to many real-world applications.

Table 14: 80 LUs invoking the Obviousness frame according to our new FrameNet+. New LUs are shown in bold. They were added using our paraphrases and human vetting.

accurate, ambiguous, apparent, apparently, audible**, axiomatic, blatant, blatantly, blurred, blurry, certainly, clarify,** clarity, clear, clearly**, confused, confusing, conspicuous, crystal-clear, dark, definite, definitely, demonstrably, discernible, distinct,** evident**, evidently, explicit, explicitly, flagrant, fuzzy, glaring, imprecise, inaccurate, lucid,** manifest**, manifestly, markedly, naturally, notable, noticeable, obscure, observable,** obvious, obviously**, opaque, openly, overt, patently, perceptible, plain, precise, prominent, self-evident,** show, show up**, significantly, soberly, specific, straightforward, strong, sure, tangible, transparent, unambiguous, unambiguously, uncertain,** unclear**, undoubtedly, unequivocal, unequivocally, unspecific, vague, viewable, visibility, visible, visibly,** visual**, vividly, woolly**

We tripled the lexical coverage of FrameNet quickly and with high precision. We used a two stage process: 1) we used rules from PPDB to automatically paraphrase FN sentences and 2) we applyed crowdsourcing to manually verify that the automatic paraphrases are of high quality. We used this process to build FrameNet+, a huge, manually-vetted extension to the current FrameNet. FrameNet+ provides over 22,000 new frame/LU mappings in a format that can be readily incorporated into existing systems. We demonstrated that the expanded resource provides a 40% improvement in lexical coverage in a practical setting.

4.4.1 Expanding FrameNet Automatically

The Paraphrase Database (PPDB) (Ganitkevitch et al., 2013) is an enormous collection of lexical, phrasal, and syntactic paraphrases. The database is released in six sizes (S to XXXL) ranging from highest precision/lowest recall to lowest average precision/highest recall. We focus on lexical (single word) paraphrases from the XL distribution, of which there are over 370K.

Our aim is to increase the type-level coverage of FN. We use the rules in PPDB along with a 5-gram Kneser-Ney smoothed language model (Heafield et al., 2013) to paraphrase FN's full frame-annotated sentences (called *fulltext*). We ignore paraphrase rules which are redundant with LUs already covered by FN. This method for automatic paraphrasing has been discussed previously by Rastogi and Van Durme (2014). However, whereas their work only discussed the idea as a hypothetical way of augmenting FN, we apply the method, vet the results, and release it as a public resource.

In total, we generate 188,061 paraphrased sentences, covering 686 frames. Table 15 shows some of the paraphrases produced.

Table 15: Example paraphrases from FrameNet's annotated full text. The bolded words are automatically proposed rewrites from PPDB.

Frame	Original	Paraphrase	Frame-annotated sentence
Quantity	amount	figure	It is not clear if this **figure** includes the munitions. . .
Expertise	expertise	specialization	. . . the technology, **specialization**, and infrastructure. . .
Labeling	called	dubbed	. . . eliminate who he **dubbed** Sheiks of sodomite. . .
Importance	significant	noteworthy	. . . assistance provided since the 1990s is **noteworthy**. . .
Mental_property	crazy	berserk	You know it's **berserk**.

4.4.2 Manual Refining with Crowdsourcing

Our automatic process produces a large number of good paraphrases, but does not address issues like word sense, and many of the paraphrased LUs alter the sentence so that it no longer evokes the intended frame. For example, PPDB proposes *free* as a paraphrase of *open*. This is a good paraphrase in the Secrecy status frame but does not hold for the Openness frame (Table 16).

Table 16: Turkers approved free as a paraphrase of open for the Secrecy status frame (rating of 4.3) but rejected it in the Openness frame (rating of 1.6).

✓	Secrecy_status
	The facilities are **open** to public scrutiny
	The facilities are **free** to public scrutiny
X	Openness
	Museum (**open** Wednesday and Friday.)
	Museum (**free** Wednesday and Friday.)

We therefore refine the automatic paraphrases manually to remove paraphrased LUs which do not evoke the same frame as the original LU. We show each sentence to three unique workers on Amazon Mechanical Turk (MTurk) and ask each to judge how well the paraphrase retains the meaning of the original phrase. We use the 5-point grading scale for paraphrase proposed by Callison-Burch (2008).

To ensure that annotators perform our task conscientiously, we embed gold-standard control sentences taken from WordNet synsets. Overall, workers were 76% accurate on our controls and showed good levels of agreement– the average correlation between two annotators' ratings was ρ = 0.49.

Figure 22 shows the distribution of Turkers' ratings for the 188K automatically paraphrased targets. In 44% of cases, the new LU was judged to retain the meaning of the original LU given the frame-specific context. These 85K sentences contain 22K unique frame/LU mappings which we are able to confidently add to FN, tripling the total number in the resource. Table 14 shows 69 new LUs added to the Obviousness frame.

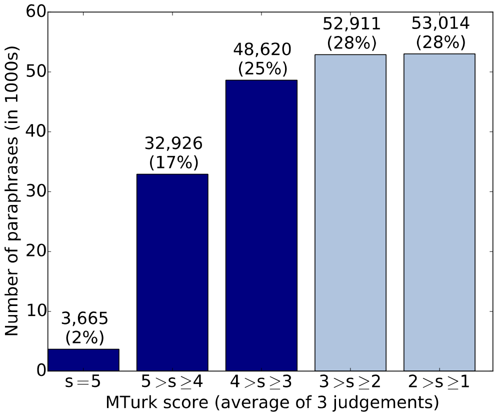


Figure 22: Distribution of MTurk ratings for paraphrased fulltext sentences. 44% received an average rating of 3, indicating the paraphrased LU was a good fit for the frame-specific context.

4.4.3 Evaluation

We aim to measure the type-level coverage improvements provided by our expanded FrameNet in a practical setting. Ideally, one would like to identify frames evoked by arbitrary sentences from natural text. To emulate this setting, we consider potentially frame-evoking LUs sampled from the New York Times. The question we ask is: does the resource contain an entry associating this LU with the frame that is actually evoked by this target?

4.4.3.1 FrameNet+

We refer to the expanded FrameNet, which contains the current FN's LUs as well as the proposed paraphrased LUs, as FrameNet+. The size and precision of FrameNet+ can be tuned by setting a threshold t and only including LU/frame mappings for which the average MTurk rating was at least t. Setting t = 0 includes all paraphrases, even those which humans judged to be incorrect, while setting t > 5 includes no paraphrases, and is equal to the current FN. Unless otherwise specified, we set t = 3. This includes all paraphrases which were judged minimally as "retaining the meaning of the original."

4.4.3.2 Sampling

LUs We consider a word to be "potentially frame-evoking" if FN+ (t = 0) contains some entry for the word, i.e. the word is either an LU in the current FN or appears in PPDB-XL as a paraphrase of some LU in the current FN. We sample 300 potentially frame-evoking word types from the New York Times: 100 each nouns, verbs, and adjectives. We take a stratified sample: within each POS, types are divided into buckets based on their frequency, and we sample uniformly from each bucket.

4.4.3.3 Annotation

For each of the potentially frame-evoking words in our sample, we have expert (non-MTurk) annotators determine the frame evoked. The annotator is given the candidate LU in the context of the New York Times sentence in which it occurred, and is shown the list of frames which are potentially evoked by this LU according to FrameNet+. The annotator then chooses which of the proposed frames fits the target, or determines that none do. We measure agreement by having two experts label each target. On average, agreement was good ($\kappa = 0.56$). In cases where they disagreed, the annotators discussed and came to a final consensus.

4.4.3.4 Results

We compute the *coverage* of a resource as the percent of targets for which the resource contained a correct LU/frame mapping. Figure 23 shows the coverage computed for the current FN compared to FN+. By including the human-vetted paraphrases, FN+ is able to return a correct LU/frame mapping for 60% of the targets in our sample, 40% more targets than were covered by the current FN. Table 17 shows some sentences covered by FN+ that are missed by the current FN.

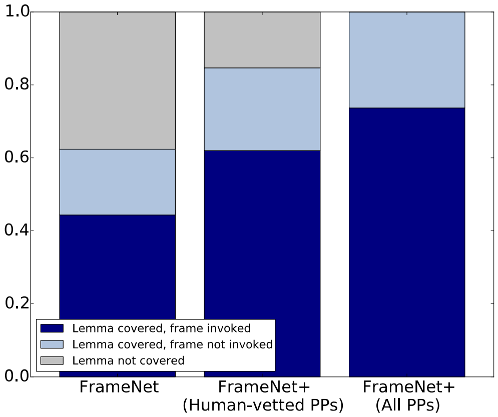


Figure 23: Number of LUs covered by the current FrameNet vs. two versions of FrameNet+: one including manually-approved paraphrases (t = 3), and one including all paraphrases (t = 0).

Table 17: Example sentences from the New York Times. The frame-invoking LUs in these sentences are not currently covered by FrameNet but are covered by the proposed FrameNet+.

LU	Frame	NYT Sentence
outsider	Indigenous_ origin	. . . I get more than my fair share because I 'm the ultimate **outsider**. . .
mini	Size	. . . a **mini** version of "The King and I " . . .
prod	Attempt_ suasion	He gently **prods** his patient to step out of his private world. . .
precious	Expensiveness	Keeping **precious** artwork safe.
sudden	Expectation	. . . on the **sudden** passing of David .

Figure 24 compares FN+'s coverage and number of LUs per frame using different paraphrase quality thresholds t. FN+ provides an average of more than 40 LUs per frame, compared to just over 10 LUs per frame in the current FN. Adding un-vetted LU paraphrases (setting t = 0) provides nearly 70 LUs per frame and offers 71% coverage.

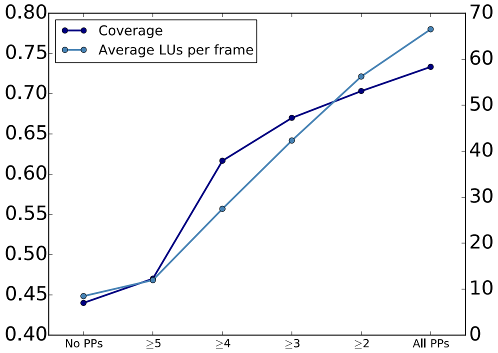


Figure 24: Overall coverage and average number of LUs per frame for varying values of t.

4.4.4 Data Release

The augmented FrameNet+ is available to download at

http://www.seas.upenn. edu/~nlp/resources/FN+.zip

The resource contains over 22K new manually-verified LU/frame pairs, making it three times larger than the currently available FrameNet. Table 18 shows the distribution of FN+'s full set of LUs by part of speech.

Table 18: Part of speech distribution for 31K LUs in FrameNet+.

Noun	12,786	Prep.	455	Conj.	14
Verb	10,862	Number	163	Wh-adv.	12
Adj.	6,195	Article	43	Particle	6
Adv.	749	Modal	22	Other	19

The release also contains 85K human-approved paraphrases of FN's fulltext. This is a huge increase over the 4K fulltext sentences currently in FN, and the new data can be easily used to retrain existing frame semantic parsers, improving their coverage at application time.

5 CONCLUSION

In this project, we devised methods to automatically extract large-volumes of paraphrases to aid in natural language understanding tasks. Our work introduced the paraphrase database (PPDB), which is now an influential and high-cited resource that has dramatically impacted research into vector embeddings for words and phrases. We advanced the state-of-the-art in data-driven paraphrasing by showing how to automatically classify paraphrase pairs with an interpretable semantics, and how paraphrase lists could be used to address traditional NLU problems like word sense induction. We produced multiple releases of PPDB, that included improvements like discriminatively re-ranked paraphrase lists with much higher correlation with human judgments of paraphrase quality. We introduced novel techniques for refining paraphrase sets so that they were applicable to specific domains. Our bilingual pivoting method allowed us to generate paraphrase databases for 23 foreign languages: Arabic, Bulgarian, Chinese, Czech, Dutch, Estonian, Finnish, French, German, Greek, Hungarian, Italian, Latvian, Lithuanian, Polish, Portuguese, Romanian, Russian, Slovak, Slovenian, and Swedish. We demonstrated the usefulness of paraphrases to semi-automatically expand manually crafted lexical-semantic resource like FrameNet, so that its coverage was tripled and little to no cost.

6 RECOMMENDATIONS

In this work, we demonstrate the effectiveness of large-scale paraphrasing for a variety of natural language understanding. There is of course, much more work to be done. We believe there is significant future work to be done in unifying word embeddings and vector space models with data-driven paraphrases that are extracted from bilingual parallel corpora, and incorporating more nuanced semantic models to allow additional types of inferences to be supported. Additionally, there are many NLU tasks that paraphrases may be able to facilitate. For instance, better question answering, or better translation for low-resource languages.

7 REFERENCES

Marianna Apidianaki and Yifan He. 2010. An algorithm for cross-lingual sense clustering tested in a MT evaluation setting. In Proceedings of the 7th International Workshop on Spoken Language Translation (IWSLT10).

Marianna Apidianaki, Emilia Verzeni, and Diana McCarthy. 2014. Semantic clustering of pivot paraphrases. In Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC 2014).

Colin Bannard and Chris Callison-Burch. 2005. Paraphrasing with bilingual parallel corpora. In Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL 05).

Islam Beltagy, Stephen Roller, Gemma Boleda, Katrin Erk, and Raymond J Mooney. 2014. Utexas: Natural language semantics using distributional semantics and probabilistic logic. In SemEval.

Roni Ben Aharon, Idan Szpektor, and Ido Dagan. 2010. Generating entailment rules from FrameNet. In ACL, pages 241–246.

Jonathan Berant, Jacob Goldberger, and Ido Dagan. 2011. Global learning of typed entailment rules. In Proceedings of ACL.

Rahul Bhagat and Deepak Ravichandran. 2008. Large scale acquisition of paraphrases for learning surface patterns. In Proceedings of ACL/HLT.

Rahul Bhagat, Patrick Pantel, Eduard H Hovy, and Marina Rey. 2007. LEDIR: An unsupervised algorithm for learning directionality of inference rules. In EMNLP-CoNLL, pages 161–170.

Johannes Bjerva, Johan Bos, Rob van der Goot, and Malvina Nissim. 2014. The meaning factory: Formal semantics for recognizing textual entailment and determining semantic similarity. In SemEval.

Thorsten Brants and Alex Franz. 2006. Web 1T 5-gram version 1.

Thorsten Brants and Alex Franz. 2009. Web 1T 5-gram, 10 European languages version 1. Linguistic Data Consortium, Philadelphia.

Aljoscha Burchardt and Anette Frank. 2006. Approaching textual entailment with LFG and FrameNet frames. In Proceedings of the Second PASCAL RTE Challenge Workshop.

Olivia Buzek, Philip Resnik, and Benjamin B Bederson. 2010. Error driven paraphrase annotation using mechanical turk. In Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon’s Mechanical Turk, pages 217– 221. Association for Computational Linguistics.

Chris Callison-Burch, Philipp Koehn, Christof Monz, and Josh Schroeder. 2009. Findings of the 2009 Workshop on Statistical Machine Translation. In Proceedings of WMT, pages 1–28, Athens, Greece, March.

Chris Callison-Burch. 2008. Syntactic constraints on paraphrases extracted from parallel corpora. In Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing, pages 196–205, Honolulu, Hawaii, October. Association for Computational Linguistics.

Tsz-Ping Chan, Chris Callison-Burch, and Benjamin Van Durme. 2011. Reranking bilingually extracted paraphrases using monolingual distributional similarity. In GEMS, pages 33–42.

David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In Proceedings of ACL.

Janara Christensen, Stephen Soderland, Oren Etzioni, et al. 2010. Semantic role labeling for open information extraction. In Proceedings of the NAACL HLT 2010 First International Workshop on Formalisms and Methodology for Learning by Reading, pages 52–60. Association for Computational Linguistics.

Kenneth Church and Patrick Hanks. 1991. Word association norms, mutual information and lexicography. Computational Linguistics, 6(1):22–29.

Trevor Cohn and Mirella Lapata. 2008. Sentence compression beyond word deletion. In Proceedings of the COLING.

Dipanjan Das and Noah A Smith. 2011. Semi-supervised frame-semantic parsing for unknown predicates. In ACL, pages 1435–1444.

Dipanjan Das and Noah A Smith. 2012. Graph-based lexicon expansion with sparsity-inducing penalties. In NAACL.

Paramveer Dhillon, Dean Foster, and Lyle Ungar. 2011. Multi-view learning of word embeddings via CCA. In NIPS.

Bill Dolan, Chris Quirk, and Chris Brockett. 2004. Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. In Proceedings of the COLING.

William Dolan, Chris Quirk, and Chris Brockett. 2004. Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. In Proceedings of the International Conference of Computational Linguistics (COLING 2004).

Joseph Dunn. 1973. A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters.

Andreas Eisele and Yu Chen. 2010. MultiUN: A multilingual corpus from united nation documents. In Proceedings of LREC, Valletta, Malta.

Michael Ellsworth and Adam Janin. 2007. Mutaphrase: Paraphrasing with framenet. In Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing, pages 143–150.

Collin F Baker, Charles J Fillmore, and John B Lowe. 1998. The Berkeley FrameNet Project. In COLING.

Manaal Faruqui, Jesse Dodge, Sujay K Jauhar, Chris Dyer, Eduard Hovy, and Noah A Smith. 2015. Retrofitting word vectors to semantic lexicons. In NAACL.

Oscar Ferrandez, Michael Ellsworth, Rafael Munoz, and Collin F Baker. 2010. Aligning FrameNet and WordNet based on semantic neighborhoods. In LREC.

Charles Fillmore and Collin F Baker. 2001. Frame semantics for text understanding. In Proceedings of WordNet and Other Lexical Resources Workshop, NAACL. Association for Computational Linguistics.

Charles Fillmore. 1982. Frame semantics. Linguistics in the morning calm.

Marco Fossati, Claudio Giuliano, and Sara Tonelli. 2013. Outsourcing FrameNet to the crowd. In ACL, pages 742–747.

William Gale, Kenneth W. Church, and David Yarowsky. 1992. Using bilingual materials to develop word sense disambiguation methods. In Proceedings of the Fourth International Conference on Theoretical and Methodological Issues in Machine Translation.

Juri Ganitkevitch and Chris Callison-Burch. 2014. The multilingual paraphrase database. In Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014), Reykjavik, Iceland, pages 4276–4283.

Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2012. Monolingual distributional similarity for text-to-text generation. In Proceedings of *SEM. Association for Computational Linguistics.

Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2013. PPDB: The paraphrase database. In NAACL-HLT, pages 758–764, Atlanta, Georgia, June.

Juri Ganitkevitch, Chris Callison-Burch, Courtney Napoles, and Benjamin Van Durme. 2011. Learning sentential paraphrases from bilingual parallel corpora for text-to-text generation. In EMNLP, pages 1168–1179.

Daniel Gildea and Daniel Jurafsky. 2002. Automatic labeling of semantic roles. Computational linguistics, 28(3):245–288.

Michelle Gregory, Liam McGrath, Eric Belanga Bell, Kelly O’Hara, and Kelly Domico. 2011. Domain independent knowledge base population from structured and unstructured data sources. In FLAIRS Conference.

Lushan Han, Abhay Kashyap, Tim Finin, James Mayfield, and Jonathan Weese. 2013. UMBC ebiquitycore: Semantic textual similarity systems. In Proceedings of the Second Joint Conference on Lexical and Computational Semantics, volume 1, pages 44– 52.

Kenneth Heafield, Ivan Pouzyrevsky, Jonathan H Clark, and Philipp Koehn. 2013. Scalable modified kneser-ney language model estimation. In ACL.

Karl Hermann, Dipanjan Das, Jason Weston, and Kuzman Ganchev. 2014. Semantic frame identification with distributed word representations. In ACL.

Yangfeng Ji and Jacob Eisenstein. 2013. Discriminative improvements to distributional sentence similarity. In EMNLP, pages 891–896.

Richard Johansson and Pierre Nugues. 2007. Using WordNet to extend FrameNet coverage. In Building Frame Semantics Resources for Scandinavian and Baltic Languages, pages 27–30. Department of Computer Science, Lund University.

Paul Kingsbury and Martha Palmer. 2002. From treebank to PropBank. In Proceedings of LREC.

Philipp Koehn and Josh Schroeder. 2007. Experiments in domain adaptation for statistical machine translation. In Proceedings of WMT, Prague, Czech Republic, June. Association for Computational Linguistics.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In MT summit, volume 5. Philipp Koehn. 2010. Statistical Machine Translation. Cambridge University Press.

Mirella Lapata and Frank Keller. 2005. Web-based models for natural language processing. ACM Transactions on Speech and Language Processing, 2(1).

Beth Levin. 1993. English verb classes and alternations: A preliminary investigation. University of Chicago press.

Dekang Lin and Patrick Pantel. 2001. Discovery of inference rules for question answering. Natural Language Engineering.

Dekang Lin, Kenneth Church, Heng Ji, Satoshi Sekine, David Yarowsky, Shane Bergsma, Kailash Patil, Emily Pitler, Rachel Lathbury, Vikram Rao, Kapil Dalwani, and Sushant Narsale. 2010. New tools for web-scale n-grams. In Proceedings of LREC.

Bill MacCartney. 2009. Natural language inference. Ph.D. thesis, Stanford University.

Prodromos Malakasiotis and Ion Androutsopoulos. 2011. A generate and rank approach to sentence paraphrasing. In EMNLP, pages 96–106.

Suresh Manandhar, Ioannis Klapaftis, Dmitriy Dligach, and Sameer Pradhan. 2010. SemEval-2010 Task 14: Word sense induction & disambiguation. In Proceedings of the Fifth International Workshop on Semantic Evaluations (SemEval-2010).

Mitchell Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of english: the Penn Treebank. Computational Linguistics, 19(2).

Diana McCarthy and Roberto Navigli. 2007. Semeval2007 task 10: English lexical substitution task. In Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007).

Oren Melamud, Ido Dagan, and Jacob Goldberger. 2015. Modeling word meaning in context with substitute vectors. In Human Language Technologies: The 2015 Annual Conference of the North American Chapter of the ACL.

Angelo Mendonca, David Andrew Graff, and Denise DiPersio. 2009. Spanish Gigaword Second Edition. Linguistic Data Consortium.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In Proceedings of NIPS.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In Workshop at ICLR.

George Miller. 1995. WordNet: a lexical database for English. Communications of the ACM, 38(11):39–41.

Behrang Mohit and Srini Narayanan. 2003. Semantic extraction with wide-coverage lexical resources. In NAACL-HLT, pages 64–66.

Ndapandula Nakashole, Gerhard Weikum, and Fabian Suchanek. 2012. PATTY: a taxonomy of relational patterns with semantic types. In Proceedings of EMNLP.

Courtney Napoles, Matt Gormley, and Benjamin Van Durme. 2012. Annotated gigaword. In Proceedings of AKBC-WEKEX 2012.

Roberto Navigli and Simone Paolo Ponzetto. 2012. BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. Artificial Intelligence, 193.

Roberto Navigli. 2009. Word sense disambiguation: A survey. ACM Computing Surveys.

Andrew Ng, Michael Jordan, and Y. Weiss. 2001. On spectral clustering: Analysis and an algorithm. Advances in Neural Information Processing Systems.

Alexis Palmer and Caroline Sporleder. 2010. Evaluating FrameNet-style semantic parsing: The role of coverage gaps in framenet. In COLING. Association for Computational Linguistics.

Martha Palmer, Hoa Trang Dang, and Christiane Fellbaum. 2001. Making fine-grained and coarse-grained sense distinctions, both manually and automatically. Natural Language Engineering.

Ellie Pavlick and Ani Nenkova. 2015. Inducing lexical style properties for paraphrase and genre differentiation. In NAACL.

Ellie Pavlick, Johan Bos, Malvina Nissim, Charley Beller, Benjamin Van Durme, and Chris CallisonBurch. 2015a. PPDB 2.0: Better paraphrase ranking, fine-grained entailment relations, word embeddings, and style classification. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL 2015).

Ellie Pavlick, Travis Wolfe, Pushpendre Rastogi, Chris Callison-Burch, Mark Drezde, and Benjamin Van Durme. 2015c. FrameNet+: Fast paraphrastic tripling of FrameNet. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL 2015), Beijing, China, July. Association for Computational Linguistics.

Marco Pennacchiotti, Diego De Cao, Roberto Basili, Danilo Croce, and Michael Roth. 2008. Automatic induction of FrameNet lexical units. In EMNLP.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. Proceedings of the Empiricial Methods in Natural Language Processing (EMNLP 2014), 12.

Chris Quirk, Chris Brockett, and William Dolan. 2004. Monolingual machine translation for paraphrase generation. In Proceedings of EMNLP.

Reinhard Rapp. 2003. Word sense discovery based on sense descriptor dissimilarity. In Proceedings of the Ninth Machine Translation Summit, pages 315–322.

Pushpendre Rastogi and Benjamin Van Durme. 2014. Augmenting FrameNet via PPDB. In Proceedings of the 2nd Workshop on Events: Definition, Detection, Coreference, and Representation. Association of Computational Linguistics.

Pushpendre Rastogi, Benjamin Van Durme, and Raman Arora. 2015. Multiview LSA: Representation learning via generalized CCA. In NAACL.

Stefan Riezler, Alexander Vasserman, Ioannis Tsochantaridis, Vibhu Mittal, and Yi Liu. 2007. Statistical machine translation for query expansion in answer retrieval. In Proceedings of the 45th Annual Meeting of the ACL.

Andrew Rosenberg and Julia Hirschberg. 2007. Vmeasure: A conditional entropy-based external cluster evaluation measure. In EMNLP-CoNLL, volume 7, pages 410–420.

Peter Rousseeuw. 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. Journal of computational and applied mathematics, 20:53–65.

Dan Shen and Mirella Lapata. 2007. Using semantic roles to improve question answering. In EMNLP-CoNLL.

Lei Shi and Rada Mihalcea. 2005. Putting pieces together: Combining FrameNet, VerbNet and WordNet for robust semantic parsing. In Computational linguistics and intelligent text processing. Springer.

Matthew Snover, Nitin Madnani, Bonnie Dorr, and Richard Schwartz. 2010. Ter-plus: paraphrase, semantic, and alignment enhancements to translation edit rate. Machine Translation, 23(2-3):117–127.

Rion Snow, Daniel Jurafsky, and Andrew Y Ng. 2004. Learning syntactic patterns for automatic hypernym discovery. In NIPS.

Rion Snow, Daniel Jurafsky, and Andrew Y. Ng. 2006. Semantic taxonomy induction from heterogenous evidence. In Proceedings of the ACL/Coling.

Ralf Steinberger, Bruno Pouliquen, Anna Widiger, Camelia Ignat, Tomaz Erjavec, Dan Tufis, and Da´niel Varga. 2006. The JRC-Acquis: A multilingual aligned parallel corpus with 20+ languages. In Proceedings of LREC, Genoa, Italy.

Md-Arafat Sultan, Steven Bethard, and Tamara Sumner. 2014a. Back to basics for monolingual alignment: Exploiting word similarity and contextual evidence. Transactions of the Association for Computational Linguistics, 2:219–230.

Md-Arafat Sultan, Steven Bethard, and Tamara Sumner. 2014b. Sentence similarity from word alignment. In SemEval.

Lin Sun and Anna Korhonen. 2011. Hierarchical verb clustering using graph factorization. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, pages 1023–1033. Association for Computational Linguistics.

Mona Talat Diab. 2003. Word sense disambiguation within a multilingual framework. Ph.D. thesis, University of Maryland.

Jorg Tiedemann. 2009. News from OPUS: A collection of multilingual parallel corpora with tools and interfaces. In Recent Advances in Natural Language Processing, volume 5.

Alfred V. Aho and Jeffrey D. Ullman. 1972. The Theory of Parsing, Translation, and Compiling. Prentice Hall.

Benjamin Van Durme and Ashwin Lall. 2010. Online generation of locality sensitive hash signatures. In Proceedings of ACL, Short Papers.

Wei Xu, Alan Ritter, Bill Dolan, Ralph Grishman, and Colin Cherry. 2012. Paraphrasing for style. In COLING.

Wei Xu, Chris Callison-Burch, and Courtney Napoles. 2015. Problems in current text simplification research: New data can help. TACL.

Xuchen Yao, Benjamin Van Durme, Chris CallisonBurch, and Peter Clark. 2013. Semi-markov phrasebased monolingual alignment. In EMNLP, pages 590–600.

Xuchen Yao, Benjamin Van Durme, and Chris CallisonBurch. 2012. Expectations of word sense in parallel corpora. In The 2012 Conference of the North American Chapter of the Association for Computational Linguistics, pages 621–625, Montre´al, Canada, June. Association for Computational Linguistics.

Mehmet-Ali Yatbaz, Enis Sert, and Deniz Yuret. 2012. Learning syntactic categories using paradigmatic representations of word context. In Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pages 940–951. Association for Computational Linguistics.

Mo Yu and Mark Dredze. 2014. Improving lexical embeddings with semantic knowledge. In ACL, volume 2, pages 545–550.

Kai Yu, Shipeng Yu, and Volker Tresp. 2005. Soft clustering on graphs. In Advances in neural information processing systems, pages 1553–1560.

Lihi Zelnik-Manor and Pietro Perona. 2004. Self-tuning spectral clustering. In Advances in neural information processing systems, pages 1601–1608.

Shiqi Zhao, Cheng Niu, Ming Zhou, Ting Liu, and Sheng Li. 2008. Combining multiple resources to improve SMT-based paraphrasing model. In Proceedings of ACL/HLT.

Liang Zhou, Chin-Yew Lin, Dragos Stefan Munteanu, and Eduard Hovy. 2006. Paraeval: Using paraphrases to evaluate summaries automatically. In Proceedings of HLT/NAACL.

APPENDIX A - Publications

2017

Word Sense Filtering Improves Embedding-Based Lexical Substitution. Anne Cocos, Marianna Apidianaki and Chris Callison-Burch. Workshop on Sense, Concept and Entity Representations and their Applications 2017

Robsut Wrod Reocginiton via semi-Character Recurrent Neural Network. Keisuke Sakaguchi, Kevin Duh, Matt Post and Benjamin Van Durme. AAAI. 2017.

Semantic Proto-Role Labeling. Adam Teichert, Adam Poliak, Benjamin Van Durme and Matthew Gormley. AAAI. 2017.

Universal Decompositional Semantics. Aaron White, Tim Viera, Drew Reisinger, Sheng Zhang, Rachel Rudinger, Keisuke Sakaguchi, Kyle Rawlins, Benjamin Van Durme. EMNLP-2016.

2016

Tense Manages to Predict Implicative Behavior in Verbs. Ellie Pavlick and Chris Callison-Burch. EMNLP-2016.

So-Called Non-Subsective Adjectives. Ellie Pavlick and Chris Callison-Burch. STARSEM-2016. Best Paper Award.

Speed-Accuracy Tradeoffs in Tagging with Variable-Order CRFs and Structured Sparsity. Tim Vieira, Ryan Cotterell and Jason Eisner. EMNLP-2016.

Learning to Prune: Pushing the Frontier of Fast and Accurate Parsing. Tim Vieira and Jason Eisner. TACL-2016.

Most babies are little and most problems are huge: Compositional Entailment in Adjective-Nouns. Ellie Pavlick and Chris Callison-Burch. ACL-2016.

Clustering Paraphrases by Word Sense. Anne Cocos and Chris Callison-Burch. NAACL-2016.

Sentential Paraphrasing as Black-Box Machine Translation. Courtney Napoles, Chris Callison-Burch, and Matt Post. NAACL-2016.

Simple PPDB: A Paraphrase Database for Simplification. Ellie Pavlick and Chris Callison-Burch. ACL-2016.

Entity recommendations on a Cold Start Knowledge Graph. Pushpendre Rastogi, Vince Lyzinski, Benjamin Van Durme. Unpublished Technical Report. Delivered to DARPA May 12, 2016.

2015

PPDB 2.0: Better paraphrase ranking, fine-grained entailment relations, word embeddings, and style classification. Ellie Pavlick, Pushpendre Rastogi, Juri Ganitkevich, Ben Van Durme, Chris Callison-Burch. ACL-2015.

Domain-Specific Paraphrase Extraction. Ellie Pavlick, Juri Ganitkevich, Tsz Ping Chan, Xuchen Yao, Ben Van Durme, Chris Callison-Burch. ACL-2015.

FrameNet+: Fast Paraphrastic Tripling of FrameNet. Ellie Pavlick, Travis Wolfe, Pushpendre

Rastogi, Chris Callison-Burch, Mark Drezde, Ben Van Durme. ACL-2015.

Script Induction as Language Modeling. Rachel Rudinger, Pushpendre Rastogi, Francis Ferraro, and Benjamin Van Durme. EMNLP-2015.

Predicate Argument Alignment using a Global Coherence Model. Travis Wolfe, Mark Dredze, and Benjamin Van Durme. NAACL. 2015.

Multiview LSA: Representation Learning via Generalized CCA. Pushpendre Rastogi, Benjamin Van Durme, and Raman Arora. NAACL. 2015.

Adding Semantics to Data-Driven Paraphrasing. Ellie Pavlick, Johan Bos, Malvina Nissim, Charley Beller, Benjamin Van Durme, and Chris Callison-Burch. ACL 2015.

2014

Extracting Lexically Divergent Paraphrases from Twitter. Wei Xu, Alan Ritter, Chris Callison-Burch, William B. Dolan and Yangfeng Ji. In TACL-2014

PARADIGM: Paraphrase Diagnostics through Grammar Matching. Jonathan Weese, Juri Ganitkevitch, and Chris Callison-Burch. EACL-2014.

The Multilingual Paraphrase Database. Juri Ganitkevitch and Chris Callison-Burch. In LREC-2014.

Augmenting FrameNet Via PPDB. Pushpendre Rastogi and Benjamin Van Durme. ACL Workshop: EVENTS. 2014.

Is the Stanford Dependency Representation Semantic? Rachel Rudinger and Benjamin Van Durme. ACL Workshop: EVENTS. 2014.

Information Extraction over Structured Data: Question Answering with Freebase. Xuchen Yao and Benjamin Van Durme. ACL 2014.

Low-Resource Semantic Role Labeling. Matthew R. Gormley and Margaret Mitchell and Benjamin Van Durme and Mark Dredze. ACL 2014.

Freebase QA: Information Extraction or Semantic Parsing? Xuchen Yao, Jonathan Berant and Benjamin Van Durme. ACL Workshop on Semantic Parsing 2014.

A Comparison of the Events and Relations Across ACE, ERE, TAC-KBP, and FrameNet Annotation Standards. Jacqueline Aguilar, Charley Beller, Paul McNamee, Benjamin Van Durme, Stephanie Strassel, Zhiyi Song and Joe Ellis. ACL Workshop: EVENTS. 2014.

Concretely Annotated Corpora. Francis Ferraro, Max Thomas, Matthew R. Gormley, Travis Wolfe, Craig Harman, and Benjamin Van Durme. NIPS Workshop on Automated Knowledge Base Construction (AKBC). 2014.

Faster (and Better) Entity Linking with Cascades. Adrian Benton, Jay Deyoung, Adam Teichert, Mark Dredze, Benjamin Van Durme, Stephen Mayhew, and Max Thomas. NIPS Workshop on Automated Knowledge Base Construction (AKBC). 2014.

2013

Semi-Markov Phrase-based Monolingual Alignment. Xuchen Yao, Ben Van Durme, Chris Callison-Burch and Peter Clark. In EMNLP-2013.

A Lightweight and High Performance Monolingual Word Aligner. Xuchen Yao, Peter Clark, Ben Van Durme and Chris Callison-Burch. In ACL-2013.

PARMA: A Predicate Argument Aligner. Travis Wolfe, Benjamin Van Durme, Mark Dredze, Nicholas Andrews, Charley Beller, Chris Callison-Burch, Jay DeYoung, Justin Snyder, Jonathan Weese, Tan Xu and Xuchen Yao. In ACL-2013.

PPDB: The Paraphrase Database. Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. In NAACL-2013.

APPENDIX B - Abstracts of Conference Papers

2017

Word Sense Filtering Improves Embedding-Based Lexical Substitution. Anne Cocos, Marianna Apidianaki and Chris Callison-Burch. Workshop on Sense, Concept and Entity Representations and their Applications 2017.

> The role of word sense disambiguation in lexical substitution has been questioned due to the high performance of vector space models which propose good substitutes without explicitly accounting for sense. We show that a filtering mechanism based on a sense inventory optimized for substitutability can improve the results of these models. Our sense inventory is constructed using a clustering method which generates paraphrase clusters that are congruent with lexical substitution annotations in a development set. The results show that lexical substitution can still benefit from senses which can improve the output of vector space paraphrase ranking models.

Robsut Wrod Reocginiton via semi-Character Recurrent Neural Network. Keisuke Sakaguchi, Kevin Duh, Matt Post and Benjamin Van Durme. AAAI-2017.

> The Cmabrigde Uinervtisy (Cambridge University) effect from the psycholinguistics literature has demonstrated a robust word processing mechanism in humans, where jumbled words (e.g. Cmabrigde / Cambridge) are recognized with little cost. Inspired by the findings from the Cmabrigde Uinervtisy effect, we propose a word recognition model based on a semi-character level recursive neural network (scRNN). In our experiments, we demonstrate that scRNN has significantly more robust performance in word spelling correction (i.e. word recognition) compared to existing spelling checkers. Furthermore, we demonstrate that the model is cognitively plausible by replicating a psycholinguistics experiment about human reading difficulty using our model.

Semantic Proto-Role Labeling. Adam Teichert, Adam Poliak, Benjamin Van Durme and Matthew Gormley. AAAI-2017.

> We present the first large-scale, corpus based verification of Dowty's seminal theory of proto-roles. Our results demonstrate both the need for and the feasibility of a property-based annotation scheme of semantic relationships, as opposed to the currently dominant notion of categorical roles.

Universal Decompositional Semantics. Aaron White, Tim Viera, Drew Reisinger, Sheng Zhang, Rachel Rudinger, Keisuke Sakaguchi, Kyle Rawlins, Benjamin Van Durme. EMNLP-2016.

> We present a framework for augmenting data sets from the Universal Dependencies project with Universal Decompositional Semantics. Where the Universal Dependencies project aims to provide a syntactic annotation standard that can be used consistently across many languages as well as a collection of corpora that use that standard, our extension has similar aims for semantic annotation. We describe results from annotating the English Universal Dependencies treebank, dealing with word senses, semantic roles, and event properties.

2016

Tense Manages to Predict Implicative Behavior in Verbs. Ellie Pavlick and Chris Callison-Burch. EMNLP-2016.

> Implicative verbs (e.g. manage) entail their compliment clauses, while non-implicative verbs (e.g. want) do not. For example, while managing to solve the problem entails solving the problem, no such inference follows from wanting to solve the problem. Differentiating between implicative and non-implicative verbs is therefore an essential component of natural language understanding, relevant to applications such as textual entailment and summarization. We present a simple method for predicting implicativeness which exploits known constraints on the tense of implicative verbs and their compliments. We show that this yields an effective, data-driven way of capturing this nuanced property in verbs.

So-Called Non-Subsective Adjectives. Ellie Pavlick and Chris Callison-Burch. STARSEM-2016. Best Paper Award.

> The interpretation of adjective-noun pairs plays a crucial role in tasks such as recognizing textual entailment. Formal semantics often places adjectives into a taxonomy which should dictate adjectives' entailment behavior when placed in adjective-noun compounds. However, we show experimentally that the behavior of subsective adjectives (e.g. red) versus non-subsective adjectives (e.g. fake) is not as cut and dry as often assumed. For example, inferences are not always symmetric: while ID is generally considered to be mutually exclusive with fake ID, fake ID is considered to entail ID. We discuss the implications of these findings for automated natural language understanding.

Speed-Accuracy Tradeoffs in Tagging with Variable-Order CRFs and Structured Sparsity. Tim Vieira, Ryan Cotterell and Jason Eisner. EMNLP-2016.

> We propose a method for learning the structure of variable-order CRFs, a more flexible variant of higher-order linear-chain CRFs. Variable-order CRFs achieve faster inference by including features for only some of the tag n-grams. Our learning method discovers the useful higher-order features at the same time as it trains their weights, by maximizing an objective that combines log-likelihood with a structured-sparsity regularizer. An active-set outer loop allows the feature set to grow as far as needed. On part-of-speech tagging in 5 randomly chosen languages from the Universal Dependencies dataset, our method of shrinking the model achieved a 2–6x speedup over a baseline, with no significant drop in accuracy.

Learning to Prune: Pushing the Frontier of Fast and Accurate Parsing. Tim Vieira and Jason Eisner. TACL-2016.

> Pruning hypotheses during dynamic programming is commonly used to speed up inference in settings such as parsing. Unlike prior work, we train a pruning policy under an objective that measures end-to-end performance: we search for a fast *and* accurate policy. This poses

a difficult machine learning problem, which we tackle with the LOLS algorithm. LOLS training must continually compute the effects of changing pruning decisions: we show how to make this efficient in the constituency parsing setting, via dynamic programming and change propagation algorithms. We find that optimizing end-to-end performance in this way leads to a better Pareto frontier—i.e., parsers which are more accurate for a given runtime.

Most babies are little and most problems are huge: Compositional Entailment in Adjective-Nouns. Ellie Pavlick and Chris Callison-Burch. ACL-2016.

We examine adjective-noun (AN) composition in the task of recognizing textual entailment (RTE). We analyze behavior of ANs in large corpora and show that, despite conventional wisdom, adjectives do not always restrict the denotation of the nouns they modify. We use natural logic to characterize the variety of entailment relations that can result from AN composition. Predicting these relations depends on context and on common-sense knowledge, making AN composition especially challenging for current RTE systems. We demonstrate the inability of current state-of-the-art systems to handle AN composition in a simplified RTE task which involves the insertion of only a single word.

Clustering Paraphrases by Word Sense. Anne Cocos and Chris Callison-Burch. NAACL-2016.

Automatically generated databases of English paraphrases have the drawback that they return a single list of paraphrases for an input word or phrase. This means that all senses of polysemous words are grouped together, unlike WordNet which partitions different senses into separate synsets. We present a new method for clustering paraphrases by word sense, and apply it to the Paraphrase Database (PPDB). We investigate the performance of hierarchical and spectral clustering algorithms, and systematically explore different ways of defining the similarity matrix that they use as input. Our method produces sense clusters that are qualitatively and quantitatively good, and that represent a substantial improvement to the PPDB resource.

Sentential Paraphrasing as Black-Box Machine Translation. Courtney Napoles, Chris Callison-Burch, and Matt Post. NAACL-2016.

We present a simple, prepackaged solution to generating paraphrases of English sentences. We use the Paraphrase Database (PPDB) for monolingual sentence rewriting and provide machine translation language packs: prepackaged, tuned models that can be downloaded and used to generate paraphrases on a standard Unix environment. The language packs can be treated as a black box or customized to specific tasks. In this demonstration, we will explain how to use the included interactive web-based tool to generate sentential paraphrases.

Simple PPDB: A Paraphrase Database for Simplification. Ellie Pavlick and Chris Callison-Burch. ACL-2016.

> We release the Simple Paraphrase Database, a subset of of the Paraphrase Database (PPDB) adapted for the task of text simplification. We train a supervised model to associate simplification scores with each phrase pair, producing rankings competitive with state-of-the-art lexical simplification models. Our new simplification database contains 4.4 million paraphrase rules, making it the largest available resource for lexical simplification.

2015

PPDB 2.0: Better paraphrase ranking, fine-grained entailment relations, word embeddings, and style classification. Ellie Pavlick, Pushpendre Rastogi, Juri Ganitkevich, Ben Van Durme, Chris Callison-Burch. ACL-2015.

> We present a new release of the Paraphrase Database. PPDB 2.0 includes a discriminatively re-ranked set of paraphrases that achieve a higher correlation with human judgments than PPDB 1.0's heuristic rankings. Each paraphrase pair in the database now also includes fine-grained entailment relations, word embedding similarities, and style annotations.

Domain-Specific Paraphrase Extraction. Ellie Pavlick, Juri Ganitkevich, Tsz Ping Chan, Xuchen Yao, Ben Van Durme, Chris Callison-Burch. ACL-2015.

> The validity of applying paraphrase rules depends on the domain of the text that they are being applied to. We develop a novel method for extracting domain-specific paraphrases. We adapt the bilingual pivoting paraphrase method to bias the training data to be more like our target domain of biology. Our best model results in higher precision while retaining complete recall, giving a 10% relative improvement in AUC.

FrameNet+: Fast Paraphrastic Tripling of FrameNet. Ellie Pavlick, Travis Wolfe, Pushpendre Rastogi, Chris Callison-Burch, Mark Drezde, Ben Van Durme. ACL-2015.

> We increase the lexical coverage of FrameNet through automatic paraphrasing. We use crowdsourcing to manually filter out bad paraphrases in order to ensure a high-precision resource. Our expanded FrameNet contains an additional 22K lexical units, a 3-fold increase over the current FrameNet, and achieves 40% better coverage when evaluated in a practical setting on New York Times data.

Script Induction as Language Modeling. Rachel Rudinger, Pushpendre Rastogi, Francis Ferraro, and Benjamin Van Durme. EMNLP-2015.

> The narrative cloze is an evaluation metric commonly used for work on automatic script induction. While prior work in this area has focused on count-based methods from distributional semantics, such as pointwise mutual information, we argue that the narrative cloze can be productively reframed as a language modeling task. By training a discriminative language model for this task, we attain improvements of up to 27 percent over prior methods on standard narrative cloze metrics.

Predicate Argument Alignment using a Global Coherence Model. Travis Wolfe, Mark Dredze, and Benjamin Van Durme. NAACL. 2015.

> We present a joint model for predicate argument alignment. We leverage multiple sources of semantic information, including temporal ordering constraints between events. These are combined in a max-margin framework to find a globally consistent view of entities and events across multiple documents, which leads to improvements over a very strong local baseline.

Multiview LSA: Representation Learning via Generalized CCA. Pushpendre Rastogi, Benjamin Van Durme, and Raman Arora. NAACL. 2015.

> Multiview LSA (MVLSA) is a generalization of Latent Semantic Analysis (LSA) that supports the fusion of arbitrary views of data and relies on Generalized Canonical Correlation Analysis (GCCA). We present an algorithm for fast approximate computation of GCCA, which when coupled with methods for handling missing values, is general enough to approximate some recent algorithms for inducing vector representations of words. Experiments across a comprehensive collection of test-sets show our approach to be competitive with the state of the art.

Adding Semantics to Data-Driven Paraphrasing. Ellie Pavlick, Johan Bos, Malvina Nissim, Charley Beller, Benjamin Van Durme, and Chris Callison-Burch. ACL 2015.

> We add an interpretable semantics to the paraphrase database (PPDB). To date, the relationship between phrase pairs in the database has been weakly defined as approximately equivalent. We show that these pairs represent a variety of relations, including directed entailment (little girl/girl) and exclusion (nobody/someone). We automatically assign semantic entailment relations to entries in PPDB using features derived from past work on discovering inference rules from text and semantic taxonomy induction. We demonstrate that our model assigns these relations with high accuracy. In a downstream RTE task, our labels rival relations from WordNet and improve the coverage of a proof-based RTE system by 17%.

2014

Extracting Lexically Divergent Paraphrases from Twitter. Wei Xu, Alan Ritter, Chris Callison-Burch, William B. Dolan and Yangfeng Ji. In TACL-2014.

> We present MULTIP (Multi-instance Learning Paraphrase Model), a new model suited to identify paraphrases within the short messages on Twitter. We jointly model paraphrase relations between word and sentence pairs and assume only sentence-level annotations during learning. Using this principled latent variable model alone, we achieve the performance competitive with a state-of-the-art method which combines a latent space model with a feature-based supervised classifier. Our model also captures lexically divergent paraphrases that differ from yet complement previous methods; combining our model with previous work significantly outperforms the state-of-the-art. In addition, we

present a novel annotation methodology that has allowed us to crowdsource a paraphrase corpus from Twitter. We make this new dataset available to the research community.

PARADIGM: Paraphrase Diagnostics through Grammar Matching. Jonathan Weese, Juri Ganitkevitch, and Chris Callison-Burch. EACL-2014.

Paraphrase evaluation is typically done either manually or through indirect, task-based evaluation. We introduce an intrinsic evaluation PARADIGM which measures the goodness of paraphrase collections that are represented using synchronous grammars. We formulate two measures that evaluate these paraphrase grammars using gold standard sentential paraphrases drawn from a monolingual parallel corpus. The first measure calculates how often a paraphrase grammar is able to synchronously parse the sentence pairs in the corpus. The second measure enumerates paraphrase rules from the monolingual parallel corpus and calculates the overlap between this reference paraphrase collection and the paraphrase resource being evaluated. We demonstrate the use of these evaluation metrics on paraphrase collections derived from three different data types: multiple translations of classic French novels, comparable sentence pairs drawn from different newspapers, and bilingual parallel corpora. We show that PARADIGM correlates with human judgments more strongly than BLEU on a task-based evaluation of paraphrase quality.

The Multilingual Paraphrase Database. Juri Ganitkevitch and Chris Callison-Burch. In LREC-2014.

WordNet has facilitated important research in natural language processing but its usefulness is somewhat limited by its relatively small coverage. The Paraphrase Database (PPDB) covers 650 times more words, but lacks the semantic structure of WordNet that would make it more directly useful for downstream tasks. We present a method for mapping words from PPDB to WordNet synsets with 89% accuracy. The mapping also lays important groundwork for incorporating WordNet's relations into PPDB so as to increase its utility for semantic reasoning in applications.

Augmenting FrameNet Via PPDB. Pushpendre Rastogi and Benjamin Van Durme. ACL Workshop: EVENTS. 2014.

FrameNet is a lexico-semantic dataset that embodies the theory of frame semantics. Like other semantic databases, FrameNet is incomplete. We augment it via the paraphrase database, PPDB, and gain a threefold increase in coverage at 65% precision.

Is the Stanford Dependency Representation Semantic? Rachel Rudinger and Benjamin Van Durme. ACL Workshop: EVENTS. 2014.

The Stanford Dependencies are a deep syntactic representation that are widely used for semantic tasks, like Recognizing Textual Entailment. But do they capture all of the semantic information a meaning representation ought to convey? This paper explores this question by investigating the feasibility of mapping Stanford dependency parses to Hobbsian Logical Form, a practical, event-theoretic semantic representation, using only a set of deterministic rules. Although we find that such a mapping is possible in a large number of cases, we also find cases for which such a mapping seems to require information beyond what the Stanford Dependencies encode. These cases shed light on the kinds of semantic information that are and are not present in the Stanford Dependencies.

Information Extraction over Structured Data: Question Answering with Freebase. Xuchen Yao and Benjamin Van Durme. ACL 2014.

> Answering natural language questions using the Freebase knowledge base has recently been explored as a platform for advancing the state of the art in open domain semantic parsing. Those efforts map questions to sophisticated meaning representations that are then attempted to be matched against viable answer candidates in the knowledge base. Here we show that relatively modest information extraction techniques, when paired with a web-scale corpus, can outperform these sophisticated approaches by roughly 34% relative gain.

Low-Resource Semantic Role Labeling. Matthew R. Gormley and Margaret Mitchell and Benjamin Van Durme and Mark Dredze. ACL 2014.

> We explore the extent to which high-resource manual annotations such as treebanks are necessary for the task of semantic role labeling (SRL). We examine how performance changes without syntactic supervision, comparing both joint and pipelined methods to induce latent syntax. This work highlights a new application of unsupervised grammar induction and demonstrates several approaches to SRL in the absence of supervised syntax. Our best models obtain competitive results in the high-resource setting and state-of-the-art results in the low resource setting, reaching 72.48% F1 averaged across languages. We release our code for this work along with a larger toolkit for specifying arbitrary graphical structure.

Freebase QA: Information Extraction or Semantic Parsing? Xuchen Yao, Jonathan Berant and Benjamin Van Durme. ACL Workshop on Semantic Parsing 2014.

> We contrast two seemingly distinct approaches to the task of question answering (QA) using Freebase: one based on information extraction techniques, the other on semantic parsing. Results over the same test-set were collected from two state-of-the-art, open-source systems, then analyzed in consultation with those systems? creators. We conclude that the differences between these technologies, both in task performance, and in how they get there, is not significant. This suggests that the semantic parsing community should target answering more compositional open-domain questions that are beyond the reach of more direct information extraction methods.

A Comparison of the Events and Relations Across ACE, ERE, TAC-KBP, and FrameNet Annotation Standards. Jacqueline Aguilar, Charley Beller, Paul McNamee, Benjamin Van Durme, Stephanie Strassel, Zhiyi Song and Joe Ellis. ACL Workshop: EVENTS. 2014.

The resurgence of effort within computational semantics has led to increased interest in various types of relation extraction and semantic parsing. While various manually annotated resources exist for enabling this work, these materials have been developed with different standards and goals in mind. In an effort to develop better general understanding across these resources, we provide a summary overview of the standards underlying ACE, ERE, TAC-KBP Slot-filling, and FrameNet.

Faster (and Better) Entity Linking with Cascades. Adrian Benton, Jay Deyoung, Adam Teichert, Mark Dredze, Benjamin Van Durme, Stephen Mayhew, and Max Thomas. NIPS Workshop on Automated Knowledge Base Construction (AKBC). 2014.

Entity linking requires ranking thousands of candidates for each query, a time-consuming process and a challenge for large scale linking. Many systems rely on prediction cascades to efficiently rank candidates. However, the design of these cascades often requires manual decisions about pruning and feature use, limiting the effectiveness of cascades. We present Slinky, a modular, flexible, fast and accurate entity linker based on prediction cascades. We adapt the web-ranking prediction cascade learning algorithm, Cronus, in order to learn cascades that are both accurate and fast. We show that by balancing between accurate and fast linking, this algorithm can produce Slinky configurations that are significantly faster and more accurate than a baseline configuration and an alternate cascade learning method with a fixed introduction of features.

2013

Semi-Markov Phrase-based Monolingual Alignment. Xuchen Yao, Ben Van Durme, Chris Callison-Burch and Peter Clark. In EMNLP-2013.

We introduce a novel discriminative model for phrase-based monolingual alignment using a semi-Markov CRF. Our model achieves state-of-the-art alignment accuracy on two phrase-based alignment datasets (RTE and paraphrase), while doing significantly better than other strong baselines in both non-identical alignment and phrase-only alignment. Additional experiments highlight the potential benefit of our alignment model to RTE, paraphrase identification and question answering, where even a naive application of our model's alignment score approaches the state of the art.

A Lightweight and High-Performance Monolingual Word Aligner. Xuchen Yao, Peter Clark, Ben Van Durme and Chris Callison-Burch. In ACL-2013.

Fast alignment is essential for many natural language tasks. But in the setting of monolingual alignment, previous work has not been able to align more than one sentence pair per second. We describe a discriminatively trained monolingual word aligner that uses a Conditional Random Field to globally decode the best alignment with features drawn from source and target sentences. Using just part-of-speech tags and WordNet as external resources, our aligner gives state-of-the-art result, while being an order-of-magnitude faster than the previous best performing system.

PARMA: A Predicate Argument Aligner. Travis Wolfe, Benjamin Van Durme, Mark Dredze, Nicholas Andrews, Charley Beller, Chris Callison-Burch, Jay DeYoung, Justin Snyder, Jonathan Weese, Tan Xu and Xuchen Yao. In ACL-2013.

We introduce PARMA, a system for cross-document, semantic predicate and argument alignment. Our system combines a number of linguistic resources familiar to researchers in areas such as recognizing textual entailment and question answering, integrating them into a simple discriminative model. PARMA achieves state of the art results on an existing and a new dataset. We suggest that previous efforts have focussed on data that is biased and too easy, and we provide a more difficult dataset based on translation data with a low baseline which we beat by 17% F1.

PPDB: The Paraphrase Database. Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. In NAACL-2013.

We present the 1.0 release of our paraphrase database, PPDB. Its English portion, PPDB:Eng, contains over 220 million paraphrase pairs, consisting of 73 million phrasal and 8 million lexical paraphrases, as well as 140 million paraphrase patterns, which capture many meaning-preserving syntactic transformations. The paraphrases are extracted from bilingual parallel corpora totaling over 100 million sentence pairs and over 2 billion English words. We also release PPDB:Spa, a collection of 196 million Spanish paraphrases. Each paraphrase pair in PPDB contains a set of associated scores, including paraphrase probabilities derived from the bitext data and a variety of monolingual distributional similarity scores computed from the Google n-grams and the Annotated Gigaword corpus. Our release includes pruning tools that allow users to determine their own precision/recall tradeoff.

LIST OF SYMBOLS, ABBREVIATIONS, AND ACRONYMS

≡	Equivalence
⊏	Forward Entailment
⊐	Reverse Entailment
ˆ	Negation
¬	Negation / Mutual Exclusion
|	Alternation
⌣	Cover
#	Independence

1C1PAR	One Cluster per Paraphrase
AAAI	Association for the Advancement of Artificial Intelligence (conference acronym)
ACL	Association for Computational Linguistics (conference)
AKBC	Automated Knowledge Base Construction (workshop acronym)
CCG	Combinatory Categorial Grammar
EACL	European chapter of the ACL (conference)
EMNLP	Empirical Methods in Natural Language Processing (conference)
FN	FrameNet
JJ	Adjective
H	Hypothesis
HGFC	Hierarchical Graph Factorization Clustering
KBP	Knowledge Base Population
LHS	Left hand side of a SCFG rule
LU	FrameNet Lexical Units
MFS	Most Frequent Sense
NLU	Natural Language Understanding
NAACL	North American chapter of the ACL (conference)
NIPS	Neural Information Processing Systems (conference)
NN	Noun
NP	Noun Phrase
POS	Part of Speech
PPDB	Paraphrase Database
SBAR	Subordinating conjunction
SCFG	Synchronous Context Free Grammar
STARSEM	Conference on Lexical and Computational Semantics
T	Text
TACL	Transactions of the Association for Computational Linguistics
VBD	Verb, past tense
VBG	Verb, gerund or present participle
VBN	Verb, past participle
VP	Verb Phrase
WSI	Word Sense Induction